IFW 2628

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:

Applicants	:	Reuven Bakalash et al.
U.S. Application No.	:	10/579,682
International Filing Date	:	November 19, 2004
Title of Invention	:	METHOD OF AND SYSTEM FOR MULTIPLE 3-D GRAPHIC PIPELINE OVER A PC BUS
Attorney Docket No.	:	142-002USAC00

Honorable Commissioner of Patents
and Trademarks
Washington, DC 20231

INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. 1.97

Sir:

In order to fulfill Applicants' continuing obligation of candor and good faith as set forth in 37 C.F.R. 1.56, Applicants submit herewith an Information Disclosure Statement prepared in accordance with 37 C.F.R Sections 1.97, 1.98 and 1.99.

The disclosures enclosed herewith are as follows:

U.S. PUBLICATIONS

NUMBER	FILING DATE	TITLE
7,289,125	February 27, 2004	GRAPHICS DEVICE CLUSTERING WITH PCI-EXPRESS
7,248,261	December 15, 2003	METHOD AND APPARATUS TO ACCELERATE RENDERING OF SHADOW EFFECTS FOR COMPUTER-GENERATED IMAGES
7,224,359	June 14, 2004	DEPTH CLAMPING SYSTEM AND METHOD IN A HARDWARE GRAPHICS PIPELINE
7,170,515	December 5, 2003	RENDERING PIPELINE
7,170,513	July 1, 2002	SYSTEM AND METHOD FOR DISPLAY LIST OCCLUSION BRANCHING

7,145,565	May 23, 2003	DEPTH BOUNDS TESTING
7,142,215	March 11, 2003	METHOD AND APPARATUS FOR PROCESSING STENCIL DATA USING A PROGRAMMABLE GRAPHICS PROCESSOR
7,130,316	April 11, 2001	SYSTEM FOR FRAME BASED AUDIO SYNCHRONIZATION AND METHOD THEREOF
7,129,909	April 25, 2003	METHOD AND SYSTEM USING COMPRESSED DISPLAY MODE LIST
7,123,266	August 23, 2002	METHOD AND APPARATUS FOR A VIDEO GRAPHICS CIRCUIT HAVING PARALLEL PIXEL PROCESSING
7,120,816	December 5, 2003	METHOD FOR TESTING SYNCHRONIZATION AND CONNECTION STATUS OF A GRAPHICS PROCESSING UNIT MODULE
7,098,922	December 22, 2003	MULTIPLE DATA BUFFERS FOR PROCESSING GRAPHICS DATA
7,095,414	June 28, 2002	BLENDING SYSTEM AND METHOD IN AN INTEGRATED COMPUTER GRAPHICS PIPELINE
7,091,971	October 25, 2002	SYSTEM, METHOD, AND APPARATUS FOR MULTI-LEVEL HIERARCHICAL Z BUFFFERING
7,081,895	April 11, 2003	SYSTEMS AND METHODS OF MULTI-PASS DATA PROCESSING
7,080,194	September 16, 2004	METHOD AND SYSTEM FOR MEMORY ACCESS ARBITRATION FOR MINIMIZING READ/WRITE TURNAROUND PENALTIES
7,075,541	August 18, 2003	ADAPTIVE LOAD BALANCING IN A MULTI-PROCESSOR GRAPHICS PROCESSING SYSTEM

7,068,278	December 11, 2003	SYNCHRONIZED GRAPHICS PROCESSING UNITS
7,068,272	May 31, 2000	SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR Z-VALUE AND STENCIL CULLING PRIOR TO RENDERING IN A COMPUTER GRAPHICS PROCESSING PIPELINE
7,064,763	June 28, 2002	SINGLE SEMICONDUCTOR GRAPHICS PLATFORM
7,053,901	December 11, 2003	SYSTEM AND METHOD FOR ACCELERATING A SPECIAL PURPOSE PROCESSOR
7,038,692	October 25, 2000	METHOD AND APPARATUS FOR PROVIDING A VERTEX CACHE
7,038,685	June 30, 2003	PROGRAMMABLE GRAPHICS PROCESSOR FOR MULTITHREADED EXECUTION OF PROGRAMS
7,038,678	May 21, 2003	DEPENDENT TEXTURE SHADOW ANTIALIASING
7,027,972	January 24, 2001	SYSTEM FOR COLLECTING AND ANALYZING GRAPHICS DATA AND METHOD THEREOF
7,023,437	August 27, 2002	SYSTEM AND METHOD FOR ACCELERATING GRAPHICS PROCESSING USING A POST-GEOMETRY DATA STREAM DURING MULTIPLE-PASS RENDERING
7,015,915	August 12, 2003	PROGRAMMING MULTIPLE CHIPS FROM A COMMAND BUFFER
7,002,588	March 19, 2003	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR BRANCHING DURING PROGRAMMABLE VERTEX PROCESSING
6,999,076	February 19, 2002	SYSTEM, METHOD AND APPARATUS FOR EARLY CULLING

6,995,767	July 31, 2003	TRILINEAR OPTIMIZATION FOR TEXTURE FILTERING
6,992,667	March 31, 2003	SINGLE SEMICONDUCTOR GRAPHICS PLATFORM SYSTEM AND METHOD WITH SKINNING, SWIZZLING AND MASKING CAPABILITIES
6,989,840	August 31, 2001	ORDER-DEPENDENT TRANSPARENCY RENDERING SYSTEM AND METHOD
6,985,152	April 23, 2004	POINT-TO-POINT BUS BRIDGING WITHOUT A BRIDGE CONTROLLER
6,982,718	November 30, 2001	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR PROGRAMMABLE FRAGMENT PROCESSING IN A GRAPHICS PIPELINE
6,980,209	June 14, 2002	METHOD AND SYSTEM FOR SCALABLE, DATAFLOW-BASED, PROGRAMMABLE PROCESSING OF GRAPHICS DATA
6,975,319	March 24, 2000	SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR CALCULATING A LEVEL OF DETAIL (LOD) DURING COMPUTER GRAPHICS PROCESSING
6,961,057	October 12, 2000	METHOD AND APPARATUS FOR MANAGING AND ACCESSING DEPTH DATA IN A COMPUTER GRAPHICS SYSTEM
6,959,110	August 13, 2001	MULTI-MODE TEXTURE COMPRESSION ALGORITHM
6,952,206	August 12, 2002	GRAPHICS APPLICATION PROGRAM INTERFACE SYSTEM AND METHOD FOR ACCELERATING GRAPHICS PROCESSING
6,947,865	February 15, 2002	METHOD AND SYSTEM FOR DYNAMIC POWER SUPPLY VOLTAGE ADJUSTMENT FOR A

		SEMICONDUCTOR INTEGRATED CIRCUIT DEVICE
6,947,047	June 10, 2002	METHOD AND SYSTEM FOR PROGRAMMABLE PIPELINED GRAPHICS PROCESSING WITH BRANCHING INSTRUCTIONS
6,940,515	December 3, 2003	USER PROGRAMMABLE PRIMITIVE ENGINE
6,938,176	October 5, 2001	METHOD AND APPARATUS FOR POWER MANAGEMENT OF GRAPHICS PROCESSORS AND SUBSYSTEMS THAT ALLOW THE SUBSYSTEMS TO RESPOND TO ACCESSES WHEN SUBSYSTEMS ARE IDLE
6,900,810	December 3, 2003	USER PROGRAMMABLE GEOMETRY ENGINE
6,894,689	September 8, 2003	OCCLUSION CULLING METHOD AND APPARATUS FOR GRAPHICS SYSTEMS
6,894,687	June 8, 2001	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR VERTEX ATTRIBUTE ALIASING IN A GRAPHICS PIPELINE
6,876,362	July 10, 2002	OMNIDIRECTIONAL SHADOW TEXTURE MAPPING
6,870,540	June 19, 2001	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR A PROGRAMMABLE PIXEL PROCESSING MODEL WITH INSTRUCTION SET
6,864,893	November 22, 2002	METHOD AND APPARATUS FOR MODIFYING DEPTH VALUES USING PIXEL PROGRAMS
6,864,893	November 22, 2002	METHOD AND APPARATUS FOR MODIFYING DEPTH VALUES USING PIXEL PROGRAMS
6,856,320	November 10, 2000	DEMAND-BASED MEMORY SYSTEM FOR GRAPHICS APPLICATIONS

6,831,652	March 24, 2000	METHOD AND SYSTEM FOR STORING GRAPHICS DATA
6,828,987	August 7, 2001	METHOD AND APPARATUS FOR PROCESSING VIDEO AND GRAPHICS DATA
6,828,980	January 10, 2003	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR Z-TEXTURE MAPPING
6,825,843	November 22, 2002	METHOD AND APPARATUS FOR LOOP AND BRANCH INSTRUCTIONS IN A PROGRAMMABLE GRAPHICS PIPELINE
6,812,927	June 18, 2002	SYSTEM AND METHOD FOR AVOIDING DEPTH CLEARS USING A STENCIL BUFFER
6,797,998	July 16, 2002	MULTI-CONFIGURATION GPU INTERFACE DEVICE
6,779,069	September 4, 2002	COMPUTER SYSTEM WITH SOURCE-SYNCHRONOUS DIGITAL LINK
6,778,189	August 24, 2001	TWO-SIDED STENCIL TESTING SYSTEM AND METHOD
6,778,181	December 7, 2001	GRAPHICS PROCESSING SYSTEM HAVING A VIRTUAL TEXTURING ARRAY
6,778,176	June 21, 2002	SEQUENCER SYSTEM AND METHOD FOR SEQUENCING GRAPHICS PROCESSING
6,774,895	February 1, 2002	SYSTEM AND METHOD FOR DEPTH CLAMPING IN A HARDWARE GRAPHICS PIPELINE
6,744,433	August 31, 2001	SYSTEM AND METHOD FOR USING AND COLLECTING INFORMATION FROM A PLURALITY OF DEPTH LAYERS

6,734,874	January 31, 2001	GRAPHICS PROCESSING UNIT WITH TRANSFORM MODULE CAPABLE OF HANDLING SCALARS AND VECTORS
6,734,861	October 16, 2000	SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR AN INTERLOCK MODULE IN A COMPUTER GRAPHICS PROCESSING PIPELINE
6,731,298	October 2, 2000	SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR Z-TEXTURE MAPPING
6,728,820	May 26, 2000	METHOD OF CONFIGURING, CONTROLLING AND ACCESSING A BRIDGE AND APPARATUS THEREFORE
6,725,457	May 17, 2000	SEMAPHORE ENHANCEMENT TO IMPROVE SYSTEM PERFORMANCE
6,724,394	June 19, 2001	PROGRAMMABLE PIXEL SHADING ARCHITECTURE
6,704,025	August 3, 2001	SYSTEM AND METHOD FOR DUAL-DEPTH SHADOW-MAPPING
6,700,583	May 14, 2001	CONFIGURABLE BUFFER FOR MULTIPASS APPLICATIONS
6,691,180	June 5, 2001	APPARATUS FOR ACCELERATING THE RENDERING OF IMAGES
6,690,372	December 5, 2000	SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR SHADOW MAPPING
6,677,953	November 8, 2001	HARDWARE VIEWPORT SYSTEM AND METHOD FOR USE IN A GRAPHICS PIPELINE
6,670,958	May 26, 2000	METHOD AND APPARATUS FOR ROUTING DATA TO MULTIPLE GRAHICS DEVICES
6,664,963	October 12, 2001	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR

		PROGRAMMABLE SHADING USING PIXEL SHADERS
6,664,963	October 12, 2001	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR PROGRAMMABLE SHADING USING PIXEL SHADERS
6,664,960	May 10, 2001	APPARATUS FOR PROCESSING NON-PLANAR VIDEO GRAPHICS PRIMITIVES AND ASSOCIATED METHOD OF OPERATION
6,662,257	May 26, 2000	MULTIPLE DEVICE BRIDGE APPARATUS AND METHOD THEREOF
6,657,635	August 31, 2000	BINNING FLUSH IN GRAPHICS DATA PROCESSING
6,650,331	September 20, 2001	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR PERFORMING A SCISSOR OPERATION IN A GRAPHICS PROCESSING FRAMEWORK EMBODIED ON A SINGLE SEMICONDUCTOR PLATFORM
6,650,330	June 21, 2002	GRAPHICS SYSTEM AND METHOD FOR PROCESSING MULTIPLE INDEPENDENT EXECUTION THREADS
6,646,639	June 19, 2001	MODIFIED METHOD AND APPARATUS FOR IMPROVED OCCLUSION CULLING IN GRAPHICS SYSTEMS
6,636,215	March 9, 2001	HARDWARE-ASSISTED Z-PYRAMID CREATION FOR HOST-BASED OCCLUSION CULLING
6,636,212	November 14, 2000	METHOD AND APPARATUS FOR DETERMINING VISIBILITY OF GROUPS OF PIXELS
6,633,296	May 26, 2000	APPARATUS FOR PROVIDING DATA TO A PLURALITY OF GRAPHICS PROCESSORS AND METHOD THEREOF

6,593,923	August 16, 2000	SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR SHADOW MAPPING
6,577,320	March 22, 1999	METHOD AND APPARATUS FOR PROCESSING MULTIPLE TYPES OF PIXEL COMPONENT REPRESENTATIONS INCLUDING PROCESSES OF PREMULTIPLICATION, POSTMULTIPLICATION, AND COLORKEYING/CHROMAKEYING
6,577,309	September 20, 2001	SYSTEM AND METHOD FOR A GRAPHICS PROCESSING FRAMEWORK EMBODIED UTILIZING A SINGLE SEMICONDUCTOR PLATFORM
6,542,971	April 23, 2001	MEMORY ACCESS SYSTEM AND METHOD EMPLOYING AN AUXILIARY BUFFER
6,535,209	November 14, 2000	DATA STREAM SPLITTING AND STORAGE IN GRAPHICS DATA PROCESSING
6,532,525	September 29, 2000	METHOD AND APPARATUS FOR ACCESSING MEMORY
6,532,013	May 31, 2000	SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR PIXEL SHADERS FOR PROGRAMMABLE SHADING
6,502,173	September 29, 2000	SYSTEM FOR ACCESSING MEMORY AND METHOD THEREFORE
6,496404	November 6, 2001	MEMORY SYSTEM FOR USE ON A CIRCUIT BOARD IN WHICH THE NUMBER OF LOADS IS MINIMIZED
6,492,987	August 27, 1998	METHOD AND APPARATUS FOR PROCESSING OBJECT ELEMENTS THAT ARE BEING RENDERED
6,477,687	June 1, 1998	METHOD OF EMBEDDING RAMS AND OTHER MACROCELLS IN THE CORE OF AN INTEGRATED CIRCUIT CHIP

6,473,089	March 2, 1998	METHOD AND APPARATUS FOR A VIDEO GRAPHICS CIRCUIT HAVING PARALLEL PIXEL PROCESSING
6,473,086	December 9, 1999	METHOD AND APPARATUS FOR GRAPHICS PROCESSING USING PARALLEL GRAPHICS PROCESSORS
6,462,737	September 20, 2001	CLIPPING SYSTEM AND METHOD FOR A GRAPHICS PROCESSING FRAMEWORK EMBODIED ON A SINGLE SEMICONDUCTOR PLATFORM
6,442,656	August 18, 1999	METHOD AND APPARATUS FOR INTERFACING MEMORY WITH A BUS
6,415,345	August 3, 1998	BUS MASTERING INTERFACE CONTROL SYSTEM FOR TRANSFERRING MULTISTREAM DATA OVER A HOST BUS
6,352,479	August 31, 1999	INTERACTIVE GAMING SERVER AND ONLINE COMMUNITY FORUM
6,333,744	March 22, 1999	GRAPHICS PIPELINE INCLUDING COMBINER STAGES
6,337,686	January 7, 1998	METHOD AND APPARATUS FOR LINE ANTI-ALIASING
6,288,418	March 19, 1999	MULTIUSE INPUT/OUTPUT CONNECTOR ARRANGEMENT FOR GRAPHICS ACCELERATOR INTEGRATED CIRCUIT
6,201,545	September 23, 1997	METHOD AND APPARATUS FOR GENERATING SUB PIXEL MASKS IN A THREE DIMENSIONAL GRAPHIC PROCESSING SYSTEM
6,188,412	August 28, 1998	METHOD AND APPARATUS FOR PERFORMING SETUP OPERATIONS IN A VIDEO GRAPHICS SYSTEM
6,184,908	April 27, 1998	METHOD AND APPARATUS FOR CO-

		PROCESSING VIDEO GRAPHICS DATA
6,181,352	March 22, 1999	GRAPHICS PIPELINE SELECTIVELY PROVIDING MULTIPLE PIXELS OR MULTIPLE TEXTURES
6,169,553	July 2, 1997	METHOD AND APPARATUS FOR RENDERING A THREE-DIMENSIONAL SCENE HAVING SHADOWING
5,909,595	August 18, 1997	METHOD OF CONTROLLING I/O ROUTING BY SETTING CONNECTING CONTEXT FOR UTILIZING I/O PROCESSING ELEMENTS WITHIN A COMPUTER SYSTEM TO PRODUCE MULTIMEDIA EFFECTS
5,758,182	August 18, 1997	DMA CONTROLLER TRANSLATES VIRTUAL I/O DEVICE ADDRESS RECEIVED DIRECTLY FROM APPLICATION PROGRAM COMMAND TO PHYSICAL I/O DEVICE ADDRESS OF I/O DEVICE ON DEVICE BUS
5,754,866	September 2, 1997	DELAYED INTERRUPTS WITH A FIFO IN AN IMPROVED INPUT/OUTPUT ARCHITECTURE
5,740,464	November 19, 1996	ARCHITECTURE FOR PROVIDING INPUT/OUTPUT OPERATIONS IN A COMPUTER SYSTEM
5,687,357	April 14, 1995	REGISTER ARRAY FOR UTILIZING BURST MODE TRANSFER ON LOCAL BUS
2007/0195099	February 21, 2006	ASSYMETRIC MULTI-GPU PROCESSING
2007/0159488	December 15, 2006	PARALLEL ARRAY ARCHITECTURE FOR A GRAPHICS PROCESSOR
2007/0159488	December 1, 2006	PARALLEL ARAY ARCHITECTURE FOR A GRAPHICS PROCESSOR
2006/0282604	May 27, 2005	METHODS AND APPARATUS FOR PROCESSING GRAPHICS DATA USING

		MULTIPLE PROCESSING CIRCUITS
2006/0271713	May 27, 2005	COMPUTING DEVICE WITH FLEXIBLY CONFIGURABLE EXPANSION SLOTS, AND METHOD OF OPERATION
2006/0268005	July 6, 2006	METHOD AND SYSTEM FOR IMPLEMENTING MULTIPLE HIGH PRECISION AND LOW PRECISION INTERPOLATORS FOR A GRAPHICS PIPELINE
2006/0248241	March 28, 2005	UNIVERSAL STORAGE BUS ADAPTOR
2006/0267987	May 24, 2005	MASTER/SLAVE GRAPHICS ADAPTER ARRANGEMENT
2006/0225061	March 31, 2005	METHOD AND APPARATUS FOR REGISTER ALLOCATION IN PRESENCE OF HARDWARE CONSTRAINTS
2006/0221087	March 14, 2006	ADAPTIVE LOAD BALANCING IN A MULTI-PROCESSOR GRAPHICS PROCESSING SYSTEM
2006/0221086	March 14, 2006	ADAPTIVE LOAD BALANCING IN A MULTI-PROCESSOR GRAPHICS PROCESSING SYSTEM
2006/0208960	March 18, 2006	DISPLAY SPECIFIC IMAGE PROCESSING IN AN INTEGRATED CIRCUIT
2006/0202941	March 9, 2005	SYSTEM AND METHOD FOR DETERMINING ILLUMINATION OF A PIXEL BY SHADOW PLANES
2006/0123142	December 6, 2004	METHOD AND APPARATUS FOR PROVIDING PEER-TO-PEER DATA TRANSFER WITHIN A COMPUTING ENVIRONMENT
2006/0120376	December 6, 2004	METHOD AND APPARATUS FOR PROVIDING PEER-TO-PEER DATA TRANSFER WITHIN A COMPUTING ENVIRONMENT

2006/0119607	January 25, 2006	REGISTER BAED QUEUING FOR TEXTURE REQUESTS
2006/0114260	January 6, 2006	PROGRAMMING MULTIPLE CHIPS FROM A COMMAND BUFFER
2006/0101218	November 11, 2004	MEMORY CONTROLLER-ADAPTIVE 1T/2T TIMING CONTROL
2006/0059494	September 16, 2005	LOAD BALANCING
2006/0055695	September 13, 2004	INCREASED SCALABILITY IN THE FRAGMENT SHADING PIPELINE
2006/0028478	October 11, 2005	POINT-TO-POINT BUS BRIDGING WITHOUT A BRIDGE CONTROLLER
2006/0005178	July 2, 2004	OPTIMIZED CHAINING OF VERTEX AND FRAGMENT PROGRAMS
2005/0275760	March 2, 2004	MODIFYING A RASTERIZED SURFACE, SUCH A BY TRIMMING
2005/0243215	May 3, 2004	FILM-MODE (3:2/2:2 PULLDOWN) DETECTOR, METHOD AND VIDEO DEVICE
2005/0265064	November 16, 2004	DATA SAMPLING CLOCK EDGE PLACEMENT TRAINING FOR HIGH SPEED GPU-MEMORY INTERFACE
2005/0259103	May 10, 2005	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR PROGRAMMABLE FRAGMENT PROCESSING
2005/0225558	April 8, 2004	TWO LEVEL CACHE MEMORY ARCHITECTURE
2005/0243096	July 6, 2005	MEMORY CONTROLLER HUB INTERFACE
2005/0237329	April 27, 2004	GPU RENDERING TO SYSTEM MEMORY
2005/0237327	April 23, 2004	POINT-TO-POINT BUS BRIDGING

		WITHOUT A BRIDGE CONTROLLER
2005/0223124	April 1, 2004	DEADLOCK AVOIDANCE IN A BUS FABRIC
2005/0206646	May 19, 2005	DISTRIBUTED RESOURCE ARCHITECTURE AND SYSTEM
2005/0195187	March 2, 2004	METHOD AND APPARATUS FOR HIERARCHICAL Z BUFFERING AND STENCILING
2005/0195186	March 2, 2004	METHOD AND APPARATUS FOR OBJECT BASED VISIBILITY CULLING
20005/0190190	February 27, 2004	GRAPHICS DEVICE CLUSTERING WITH PCI-EXPRESS
2005/0162437	January 23, 2004	METHOD AND APPARATUS FOR GRAPHICS PROCESSING USING STATE AND SHADER MANAGEMENT
2005/0081115	September 26, 2003	METHOD AND APPARATUS FOR MONITORING AND RESETTING A CO-PROCESSOR
2005/0041031	August 18, 2003	ADAPTIVE LOAD BALANCING IN A MULTI-PROCESSOR GRAPHICS PROCESSING SYSTEM
2004/0210788	December 5, 2003	METHOD FOR TESTING SYNCHRONIZATION AND CONNECTION STATUS OF A GRAPHICS PROCESSING UNIT MODULE
2004/0207618	December 8, 2003	METHOD FOR SYNCHRONIZING GRAPHICS PROCESSING UNITS
2004/0179019	March 12, 2003	DOUBLE-BUFFERING OF PIXEL DATA USING COPY-ON-WRITE SEMANTICS
2004/0169651	May 23, 2003	DEPTH BOUNDS TESTING
2004/0153778	April 2, 2003	METHOD, SYSTEM AND SOFTWARE FOR CONFIGURING A GRAPHICS PROCESSING COMMUNICATION

		MODE
2004/0036159	August 23, 2002	INTEGRATED CIRCUIT HAVING MEMORY DISPOSED THEREON AND METHOD OF MAKING THEREOF
2004/0012600	March 21, 2003	SCALABLE HIGH PERFORMANCE 3D GRAPHICS
2003/0212735	May 13, 2002	METHOD AND APPARATUS FOR PROVIDING AN INTEGRATED NETWORK OF PROCESSORS
2003/0189565	March 31, 2003	SINGLE SEMICONDUCTOR GRAPHICS PLATFORM SYSTEM AND METHOD WITH SKINNING, SWIZZLING AND MASKING CAPABILITIES
2003/0179220	March 20, 2002	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR GENERATING A SHADER PROGRAM
2003/0151606	October 25, 2002	SYSTEM, METHOD AND APPARATUS FOR MULTI-LEVEL HIERARCHICAL Z BUFFERING
2003/0128197	January 4, 2002	PORTABLE DEVICE FOR PROVIDING DUAL DISPLAY AND METHOD THEREOF
2003/0112246	June 28, 2002	BLENDING SYTEM AND METHOD IN AN INTEGRATED COMPUTER GRAPHICS PIPELINE
2003/0112245	June 28, 2002	SINGLE SEMICONDUCTOR GRAPHICS PLATFORM
2003/0103054	July 17, 2002	INTEGRATED GRAPHICS PROCESSING UNIT WITH ANTIALIASING
2003/00080959	February 19, 2002	SYSTEM, METHOD AND APPARATUS FOR EARLY CULLING
2003/0038808	June 21, 2002	METHOD, APPARATUS AND ARTICLE OF MANUFACTURE FOR A SEQUENCER IN A TRANSFORM/LIGHTING MODULE

		CAPABLE OF PROCESSING MULTIPLE INDEPENDENT EXECUTION THREADS
2003/00034975	October 26, 2001	LIGHTING SYSTEM AND METHOD FOR A GRAPHICS PROCESSOR
2003/00020720	June 21, 2002	METHOD, APPARATUS AND ARTICLE OF MANUFACTURE FOR A SEQUENCER IN A TRANSFORM/LIGHTING MODULE CAPABLE OF PROCESSING MULTIPLE INDEPENDENT EXECUTION THREADS
2002/0196259	June 28, 2002	SINGLE SEMICONDUCTOR GRAPHICS PLATFORM WITH BLENDING AND FOG CAPABILITIES
2002/0180740	June 28, 2002	CLIPPING SYSTEM AND METHOD FOR A SINGLE GRAPHICS SEMICONDUCTOR PLATFORM
2002/0118308	February 27, 2001	INTEGRATED SINGLE AND DUAL TELEVISION TUNER HAVING IMPROVISED FINE TUNING
2002/0085007	June 28, 2001	GRAPHICS SYSTEM CONFIGURED TO PARALLEL-PROCESS GRAPHICS DATA USING MULTIPLE PIPELINES
2002/0015055	June 26, 2001	METHOD AND SYSTEM FOR PRESENTING THREE-DIMENSIONAL COMPUTER GRAPHICS IMAGES USING MULTIPLE GRAPHICS PROCESSING UNITS
2002/0029556	June 5, 2001	APPARATUS FOR ACCELERATING THE RENDERING OF IMAGES

PowerPoint Presentation entitled "Go Multiple" by Dennis Yang, Conference Platform, 10 pages, 2007.

Scientific publication entitled "Chromium: A Stream-Processing Framework for Interactive Rendering on Clusters" by Greg Humphreys et al., Stanford University et al., 10 pages, 2007.

Scientific publication entitled "HYBRID SORT-FIRST AND SORT-LAST PARALLEL RENDERING WITH A CLUSTER OF PCs" by Rudrajit Samanta et al., Princeton University, 12 pages, c. 2000.

ABSTRACTS OF DISCLOSURE

U.S. Patent No. 7,289,125 by Diard et al. discloses a bridge associated with a broadcast aperture which facilitates the transfer of rendering commands and data between a processor and multiple graphics devices. The bridge receives data written by the processor to the broadcast aperture and forwards it to multiple graphics devices, eliminating the need for the processor to perform duplicative write operations. During system initialization, a broadcast aperture is allocated to the bridge in address space based on an aperture size value set using a system configuration utility and stored in system configuration memory. A graphics driver activates the broadcast aperture by sending unicast aperture parameters associated with the multiple graphics devices to the bridge via a bridge driver. Upon activating the broadcast aperture, multiple graphics devices can be operated in parallel to improve rendering performance. Parallel rendering techniques include split-frame, alternate frame, and combined split- and alternate frame rendering.

U.S. Patent No. 7,248,261 to Hakura discloses a method and system that provides for accelerating the generation of graphical images that include shadow effects by, for example, reducing the amount of data transmitted and/or stored necessary to render graphics based on stencil shadow volumes. In one embodiment, an exemplary apparatus is configured to render shadows using stencil shadow volumes. The apparatus includes a memory to store a degree of shadowing for each sample. A co-processor, which is coupled to the memory, is configured to generate an indicator that represents a common degree of shadowing associated with the subset of samples. In some cases, the apparatus includes a graphics processing unit ("GPU"), which is coupled to the co-processor, that is configured to render one or more shadows for a computer-generated image based on the indicator.

U.S. Patent No. 7,224,359 to Papakipos et al. discloses a system, method and computer program product for depth clamping in a hardware graphics pipeline. Initially, a depth value is identified. It is then determined as to whether a hardware graphics pipeline is operating in a depth clamping mode. If the hardware graphics pipeline is operating in the depth clamping mode, the depth value is clamped within a predetermined range utilizing the hardware graphics pipeline.

U.S. Patent No. 7,170,515 to Zhu et al. discloses a rendering pipeline system for a computer environment which uses screen space tiling (SST) to eliminate the memory bandwidth

bottleneck due to frame buffer access and performs screen space tiling efficiently, while avoiding the breaking up of primitives. The system also reduces the buffering size required by SST. High quality, full-scene anti-aliasing is easily achieved because only the on-chip multi-sample memory corresponding to a single tile of the screen is needed. The invention uses a double-z scheme that decouples the scan conversion/depth-buffer processing from the more general rasterization and shading processing through a scan/z engine. The scan/z engine externally appears as a fragment generator but internally resolves visibility and allows the rest of the rendering pipeline to perform setup for only visible primitives and shade only visible fragments. The resulting reduced raster/shading requirements can lead to reduced hardware costs because one can process all parameters with generic parameter computing units instead of with dedicated parameter computing units. The invention processes both opaque and transparent geometries.

U.S. Patent No. 7,170,513 to Voorhies et al. discloses a system and method for conditional branching in a hardware graphics pipeline. Initially, a plurality of graphics commands is received. Condition data is then affected based on at least some of the graphics commands utilizing the hardware graphics pipeline. At least one of the graphics commands is then conditionally skipping based on the condition data in response to another graphics command utilizing the hardware graphics pipeline.

U.S. Patent No. 7,145,565 to Everitt et al. discloses lights that can be conservatively bounded within a depth range. When image pixels are outside of a light's depth range, an associated volume fragment does not have to be rendered. Depth bounds registers can be used to store minimum and maximum depth values for a light. As graphics hardware processes volume fragments overlapping the image, the image's depth values are compared with the values in the depth bounds register. If the image's depth is outside of the depth range for the light, stencil buffer and illumination operations for this volume fragment are bypassed. This optimization can be performed on a per-pixel basis, or simultaneously on a group of adjacent pixels. The depth bounds are calculated from the light, or from the intersection of the volume with one or more other features. A rendering application uses API functions to set the depth bounds for each light and to activate depth bounds checking.

U.S. Patent No. 7,142,215 to Papakipos et al. discloses a graphics data-processing pipeline, including a geometry processor and a fragment processor. The graphics data-processing pipeline being configured to render stencil data and to output the stencil data in a format compatible with input to the fragment processor. An output of the graphics data-processing pipeline is written to local memory and the output is subsequently read using the fragment processor without host processor intervening usage to format the stencil data or process the stencil data.

U.S. Patent No. 7,130,316 to Kovacevic discloses a system and method for synchronizing the presentation of audio data with video data. Audio transport packets are received through a demultiplexer from a multimedia transport stream. A transport stream synchronization manager is used to lock a system time clock, local to the demultiplexer, to a program clock reference provided through the multimedia transport stream. Presentation time stamps are provided with the audio transport packets to indicate when decoded audio data is to be output. A packetized elementary stream synchronization manager maintains synchronization by adding or dropping audio packets from the audio transport packets. If the packetized elementary stream manager is

unable to acquire synchronization it must defer synchronization back to the transport stream synchronization manager. Otherwise, processed audio packets are passed to an elementary stream synchronization manager that attempts to synchronize the delivery of audio data through a sample rate conversion of the audio data. If the elementary synchronization manager is unable to acquire synchronization, it defers synchronization to the packetized elementary stream synchronization manager.

U.S. Patent No. 7,129,909 to Dong et al. discloses a method and system using a compressed display mode list. In particular, the compressed display mode list includes a plurality of data representing the display modes. The data is formatted according to a plurality of compression format rules. The compression format rules reduce and minimize the size of the compressed display mode list. A driver controls a graphical processing unit that renders an image for displaying on a display device according to a selected display mode from the compressed display mode list. Moreover, a computer-readable medium can store the compressed display mode list.

U.S. Patent No. 7,123,266 to Wei et al. discloses a method and apparatus for parallel processing of pixel information within a video graphics circuit which is accomplished when the video graphics circuit includes a set-up engine, an edgewalker circuit, a span processing circuit, and a plurality of pixel processing circuits. The set-up engine receives vertex information and produces object-element information therefrom. The object-element information is provided to the edgewalker circuit, which in turn produces span definition information. The span definition information identifies the starting pixel of a span and the starting pixel parameters. The span information is received by the processing circuit and converted into a plurality of pixel parameters. The plurality of pixel parameters are provided to the plurality of pixel processing circuits wherein each of the plurality of pixel processing circuits processes corresponding pixel parameters to produce pixel information in accordance with the information provided by the processing circuit.

U.S. Patent No. 7,120,816 to Williams et al. discloses a method for testing synchronization between a first graphics processing unit coupled to a second graphics processing unit. The method includes detecting whether an incoming synchronization signal has been received, determining whether the incoming synchronization signal is received from one of the first graphics processing unit, the second graphics processing unit and an external synchronization signal, and indicating on a control panel one of a first and second synchronization input/output ports on one of the first graphics processing unit and the second graphics processing unit as an input port and the other one of the first and second synchronization input/output ports as an output port, if the incoming synchronization signal is received from the one of the first graphics processing unit and the second graphics processing unit.

U.S. Patent No. 7,098,922 to Bastos et al. discloses a multiple output buffers supported in a graphics processor. Each output buffer has a unique identifier and may include data represented in a variety of fixed and floating-point formats (8-bit, 16-bit, 32-bit, 64-bit and higher). A fragment program executed by the graphics processor can access (read or write any of the output buffers. Each of the output buffers may be read from and used to process graphics data by an execution pipeline within the graphics processor. Likewise, each output buffer may be written to

by the graphics processor, storing graphics data such as lighting parameters, indices, color, and depth.

U.S. Patent No. 7,095,414 to Lindholm et al. discloses a system and method for a hardware implementation of a blending technique during graphics processing in a graphics pipeline. During processing in the pipeline, a plurality of matrices and a plurality of weight values are received. Also received is vertex data to be processed. A sum of a plurality of products may then be calculated by the multiplication of the vertex data, one of the matrices, and at least one of the weights.

U.S. Patent No. 7,091,971 to Morein discloses a secondary representative Z value memory which includes a reduced-resolution representation of a primary representative Z value memory. Upon updating of a block of the primary representative Z value memory, one or more corresponding values are calculated for updating the reduced-resolution representation.

U.S. Patent No. 7,081,895 to Papakipos et al. discloses a method and apparatus for graphics processing. More particularly, a graphics processing subsystem capable of multi-pass graphics data processing is described. The graphics processing subsystem includes a geometry processor and a fragment processor, where output from the fragment processor is input compatible with the geometry processor. Data produced in a pass through a graphics data-processing pipeline including the fragment processor and geometry processor may be used as an input to processing during a subsequent pass. Data read from a texture map may be used to define or modify data, including vertex data, being processed in the geometry processor or the fragment processor.

U.S. Patent No. 7,080,194 to Van Dyke discloses a method and system for arbitrating among memory access commands from clients seeking access to a DRAM or other memory, and an arbiter for use in implementing such method or system. When arbitrating among competing commands that include at least one command of the same read/write type as the current command, the arbiter selects a command of the same read/write type as the current command. In a wait mode, when arbitrating among a set of the commands that includes no command of the same read/write type as the current command, the arbiter prevents each command in the set from reaching the memory. Preferably, after operating in the wait mode for a limited time, the arbiter enters another arbitration mode in which it can select a command of the opposite read/write type as the current command. Preferably, the arbiter is implemented to be operable in any of multiple operating modes. For example, it can have separately programmable wait times for "read to write" and "write to read" situations. Preferably, the arbiter monitors for occurrence of potential page fault conditions.

U.S. Patent No. 7,075,541 to Diard discloses systems and methods for balancing a load among multiple graphics processors that render different portions of a frame. A display area is partitioned into portions for each of two (or more) graphics processors. The graphics processors render their respective portions of a frame and return feedback data indicating completion of the rendering. Based on the feedback data, an imbalance can be detected between respective loads of two of the graphics processors. In the event that an imbalance exists, the display area is re-partitioned to increase a size of the portion assigned to the less heavily loaded processor and to decrease a size of the portion assigned to the more heavily loaded processor.

U.S. Patent No. 7,068,278 to Williams et al. discloses a graphics processing unit, which includes a clock generator configured to generate a clock signal and a controller coupled to the clock generator. The controller is configured to receive the clock signal, compare the clock signal with a synchronization signal to generate a timing signal, and transmit the timing signal to a second graphics processing unit connected to the graphics processing unit.

U.S. Patent No. 7,068,272 to Voorhies et al. discloses a system, method and article of manufacture for early Z-value based culling prior to pixel rendering in a graphics pipeline. In initial stages of processing, Z-value culling is performed on at least one pixel. Thereafter, the pixel is conditionally rendered. Whether the pixel is rendered or not is conditioned on results of the Z-value culling. By culling, or removing, the pixels that do not meet certain criteria prior to rendering, much processing is avoided in the rendering portion of the graphics pipeline.

U.S. Patent No. 7,064,763 to Lindholm et al. discloses a graphics pipeline system and method for graphics processing. Such system includes a transform module positioned on a single semiconductor platform for transforming graphics data from object space to screen space. Coupled to the transform module is a lighting module which is positioned on the single semiconductor platform for lighting the graphics data. Also included is a rasterizer coupled to the lighting module and positioned on the single semiconductor platform for rendering the graphics data.

U.S. Patent No. 7,053,901 to Huang et al. discloses a system and method for accelerating at least one special purpose processor, such as a GPU, or a driver managing a special purpose processor, by using at least one co-processor. Advantageously, embodiments of the invention are fault-tolerant in that the at least one GPU or other special purpose processor is able to execute all computations, although perhaps at a lower level of performance, if the at least one co-processor is rendered inoperable. The co-processor may also be used selectively, based on performance considerations.

U.S. Patent No. 7,038,692 to Priem et al. discloses a method for caching data defining vertices of a polygon to be displayed by an input/output display device including the steps of providing an index by a vertex for which data is to be cached, storing data defining attributes of a polygon at a vertex in a cache under the index provided, issuing a command signifying a polygon to be manipulated by indicating indices of the vertices of the polygon for which data is cached.

U.S. Patent No. 7,038,685 to Lindholm discloses a programmable graphics processor for multithreaded execution of program instructions including a thread control unit. The programmable graphics processor is programmed with program instructions for processing primitive, pixel and vertex data. The thread control unit has a thread storage resource including locations allocated to store thread state data associated with samples of two or more types. Sample types include primitive, pixel and vertex. A number of threads allocated to processing a sample type may be dynamically modified.

U.S. Patent No. 7,038,678 to Bunnell discloses antialiasing shadows using a programmable graphics processor. Shadows are antialiased using dependent texture mapping to displace shadow map coordinates. A jitter texture is applied to an object in screen space using

non-perspective corrected jitter texture coordinates. The jitter texture coordinates are used to read texture coordinate displacements stored as the jitter texture. The texture coordinate displacements are combined with the shadow, map coordinates to generate displaced shadow map coordinates. The displaced shadow map coordinates are used to read depth values stored as the shadow map. The depth values read from the shadow map are compared with corresponding depth values of the object in light source coordinate space to determine whether each fragment within the object is either "in shadow" or "out of shadow".

U.S. Patent No. 7,027,972 to Lee discloses a system and method for performing a hardware performance analysis of graphics hardware and an application program. An application program generates a set of function calls. The function calls are translated to a native command set. The native command set is stored within a database. Software simulations and hardware emulations are used to compare the stored native command set data to a hardware architectural description of the graphics hardware. Data collected from the simulations are used to provide a performance model from which the performance of a graphics hardware executing commands for the application program can be determined.

U.S. Patent No. 7,023,437 to Voorhies et al. discloses a system and method for accelerating graphics processing utilizing multiple-pass rendering. Initially, geometry operations are performed on graphics data, and the graphics data is stored in memory. During a first rendering pass, various operations take place. For example, the graphics data is read from the memory, and the graphics data is rasterized. Moreover, first z-culling operations are performed utilizing the graphics data. Such first z-culling operations maintain a first occlusion image. During a second rendering pass, the graphics data is read from memory. Still yet, the graphics data is rasterized and second z-culling operations are performed utilizing the graphics data and the first occlusion image. Moreover, visibility operations are performed utilizing the graphics data and a second occlusion image. Raster-processor operations are also performed utilizing the graphics data, during the second rendering pass.

U.S. Patent No. 7,015,915 to Diard discloses a CPU for selectively programming one or more graphics devices by writing a control command to the command buffer that designates a subset of graphics devices to execute subsequent commands. Graphics devices not designated by the control command will ignore the subsequent commands until re-enabled by the CPU. The non-designated graphics devices will continue to read from the command buffer to maintain synchronization. Subsequent control commands can designate different subsets of graphics devices to execute further subsequent commands. Graphics devices include graphics processing units and graphics coprocessors. A unique identifier is associated with each of the graphics devices. The control command designates a subset of graphics devices according to their respective unique identifiers. The control command includes a number of bits. Each bit is associated with one of the unique identifiers and designates the inclusion of one of the graphics devices in the first subset of graphics devices.

U.S. Patent No. 7,002,588 to Lindholm et al. discloses a system, method and computer program product for branching during graphics processing. Initially, a first operation is performed on data. In response to the first operation, a branching operation is performed to a second operation. The first operation and the second operation are associated with instructions selected from a predetermined instruction set.

U.S. Patent No. 6,999,076 to Morein discloses a method of graphics processing which includes determining a non-depth conditional status and an occlusion status of a fragment. Such a method may be used in culling occluded fragments before expending resources such as processing cycles and memory bus usage. In one example, a scratchpad stores depth values of robust fragments and is used for occlusion testing. Graphics architectures, and methods that include use of representative Z values, are also disclosed.

U.S. Patent No. 6,995,767 to Donovan et al. discloses trilinear optimization is a technique to reduce the number of texture samples used to determine a texture value associated with a graphics fragment. Bilinear interpolations replace some trilinear interpolations, thereby reducing the number of texture samples read and simplifying the filter computation. A programmable trilinear slope is used to control replacement of a trilinear computation with a bilinear computation, permitting a user to determine a balance between improved texture map performance and texture filtering quality.

U.S. Patent No. 6,992,667 to Lindholm et al. discloses a graphics hardware system and method for graphics processing. Such system includes a transform module positioned on a single semiconductor platform for transforming graphics data. Coupled to the transform module is a lighting module which is positioned on the single semiconductor platform for lighting the graphics data. Also included is a rasterizer coupled to the lighting module and positioned on the single semiconductor platform for rendering the graphics data. As an option, the graphics hardware system may further be equipped with skinning, swizzling and masking capabilities.

U.S. Patent No. 6,989,840 to Everitt et al. discloses a system, method and computer program product for transparency rendering in a graphics pipeline. Initially, colored-transparency information is collected from a plurality of depth layers (i.e. colored-transparency layers, etc.) in a scene to be rendered. The collected colored-transparency information is then stored in memory. The colored-transparency information from the depth layers may then be blended in a predetermined order.

U.S. Patent No. 6,985,152 to Rubinstein et al. discloses a computer system which includes an integrated graphics subsystem and a graphics connector for attaching either an auxiliary graphics subsystem or a loopback card. A first bus connection communicates data from the computer system to the integrated graphics subsystem. With a loopback card in place, data travels from the integrated graphics subsystem back to the computer system via a second bus connection. When the auxiliary graphics subsystem is attached, the integrated graphics subsystem operates in a data forwarding mode. Data is communicated to the integrated graphics subsystem via the first bus connection. The integrated graphics subsystem then forwards data to the auxiliary graphics subsystem. A portion of the second bus connection communicates data from the auxiliary graphics subsystem back to the computer system. The auxiliary graphics subsystem communicates display information back to the integrated graphics subsystem, where it is used to control a display device.

U.S. Patent No. 6,982,718 to Kilgard et al. discloses a system, method and computer program product for programmable processing of fragment data in a computer hardware graphics pipeline. Initially, fragment data is received in a hardware graphics pipeline. It is then

determined whether the hardware graphics pipeline is operating in a programmable mode. If it is determined that the hardware graphics pipeline is operating in the programmable mode, programmable operations are performed on the fragment data in order to generate output. The programmable operations are performed in a manner/sequence specified in a graphics application program interface. If it is determined that the hardware graphics pipeline is not operating in the programmable mode, standard graphics application program interface (API) operations are performed on the fragment data in order to generate output.

U.S. Patent No. 6,980,209 to Donham et al. discloses a scalable pipelined pixel shader that processes packets of data and preserves the format of each packet at each processing stage. Each packet is an ordered array of data values, at least one of which is an instruction pointer. Each member of the ordered array can be indicative of any type of data. As a packet progresses through the pixel shader during processing, each member of the ordered array can be replaced by a sequence of data values indicative of different types of data (e.g., an address of a texel, a texel, or a partially or fully processed color value). Information required for the pixel shader to process each packet is contained in the packet, and thus the pixel shader is scalable in the sense that it can be implemented in modular fashion to include any number of identical pipelined processing stages and can execute the same program regardless of the number of stages. Preferably, each processing stage is itself scalable, can be implemented to include an arbitrary number of identical pipelined instruction execution stages known as microblenders, and can execute the same program regardless of the number of microblenders. The current value of the instruction pointer (IP) in a packet determines the next instruction to be executed on the data contained in the packet. Any processing unit can change the instruction that will be executed by a subsequent processing unit by modifying the IP (and/or condition codes) of a packet that it asserts to the subsequent processing unit. Other aspects of the invention include graphics processors (each including a pixel shader configured in accordance with the invention), methods and systems for generating packets of data for processing in accordance with the invention, and methods for pipelined processing of packets of data.

U.S. Patent No. 6,975,319 to Donovan et al. discloses a system, method and article of manufacture for calculating a level of detail (LOD) value for use during computer graphics processing. First, a plurality of geometrically arranged coordinates is identified. A distance value is computed based on the geometrically arranged coordinates. A LOD value is then calculated using the distance value for use during computer graphics processing. In one embodiment, a derivative value is estimated based on the geometrically arranged coordinates, and the distance value is computed based on the derivative value.

U.S. Patent No. 6,961,057 to Van Dyke et al. discloses a computer graphics system provides for processing image data, including Z data for use in displaying three-dimensional images on a display unit. The system includes: a depth buffer providing for temporary storage of Z data; and a graphics processing unit having a graphics engine for generating image data including Z data, and a memory interface unit communicatively coupled to the graphics engine and communicatively coupled to the depth buffer via a depth buffer interface. The graphics processing unit is operative to compress at least a portion of the generated Z data, to write the compressed portion of Z data to the depth buffer via the depth buffer interface in a compressed format, to read portions of compressed Z data from the depth buffer via the depth buffer interface, and to decompress the compressed Z data read from the buffer. An advantage of the

present invention is that effective Z data bandwidth through the depth buffer interface is maximized in order to facilitate fast depth buffer access operations.

U.S. Patent No. 6,959,110 to Danskin et al. discloses a multi-mode texture compression algorithm for effective compression and decompression texture data during graphics processing. Initially, a request is sent to memory for compressed texture data. Such compressed texture data is then received from the memory in response to the request. At least one of a plurality of compression algorithms associated with the compressed texture data is subsequently identified. Thereafter, the compressed texture data is decompressed in accordance with the identified compression algorithm.

U.S. Patent No. 6,952,206 to Craighead discloses a system, method and computer program product for accelerating graphics processing utilizing a graphics application program interface. Initially, graphics data is processed in a graphics system with components including a central processing unit, a geometry processing module, and a pixel processing module. In use, the graphics application program interface accepts one or more first occlusion queries followed by a second occlusion query. The second occlusion query is at least partially processed by the graphics system before a final result of any one of the first occlusion queries is computed by the graphics system.

U.S. Patent No. 6,947,865 to Mimberg et al. discloses a processor power supply voltage controller. The controller includes a temperature sensor configured to sense a temperature of a processor and generate a temperature signal in accordance therewith. A regulator is coupled to provide a power supply voltage to the processor. The regulator is coupled to receive the temperature signal and control the power supply voltage to maintain a substantially stable crosstalk level within the processor.

U.S. Patent No. 6,947,047 to Moy et al. discloses a programmable, pipelined graphics processor (e.g., a vertex processor) having at least two processing pipelines, a graphics processing system including such a processor, and a pipelined graphics data processing method allowing parallel processing and also handling branching instructions and preventing conflicts among pipelines. Preferably, each pipeline processes data in accordance with a program including by executing branch instructions, and the processor is operable in any one of a parallel processing mode in which at least two data values to be processed in parallel in accordance with the same program are launched simultaneously into multiple pipelines, and a serialized mode in which only one pipeline at a time receives input data values to be processed in accordance with the program (and operation of each other pipeline is frozen). During parallel processing mode operation, mode control circuitry recognizes and resolves branch instructions to be executed (before processing of data in accordance with each branch instruction starts) and causes the processor to operate in the serialized mode when (and preferably only for as long as) necessary to prevent any conflict between the pipelines due to branching. In other embodiments, the processor is operable in any one of a parallel processing mode and a limited serialized mode in which operation of each of a sequence of pipelines (or pipeline sets) pauses for a limited number of clock cycles. The processor enters the limited serialized mode in response to detecting a conflict-causing instruction that could cause a conflict between resources shared by the pipelines during parallel processing mode operation.

U.S. Patent No. 6,940,515 to Moreton et al. discloses a fixed function engine and method for processing a set of primitive commands. One embodiment of the fixed function engine includes a means for receiving one or more primitive commands, where each such primitive command includes information for processing vertex data using a user-developed program or subroutine. The fixed function engine also includes a means for determining a set of related primitive commands from the received primitive commands and a means for identifying a first primitive command to process from that set. In addition, the fixed function engine includes a means for transmitting a first program command, which is related to the first primitive command, to a processing engine for processing.

U.S. Patent No. 6,938,176 to Alben et al. discloses a graphics processing device implementing a set of techniques for power management, preferably at both a subsystem level and a device level, and preferably including peak power management, a system including a graphics processing device that implements such a set of techniques for power management, and the power management methods performed by such a device or system. In preferred embodiments, the device includes at least two subsystems and hardware mechanisms that automatically seek the lowest power state for the device that does not impact performance of the device or of a system that includes the device. Preferably, the device includes a control unit operable in any selected one of multiple power management modes, and system software can intervene to cause the control unit to operate in any of these modes.

U.S. Patent No. to Moreton et al. discloses a programmable geometry engine. One embodiment of the programmable geometry engine includes a programmable primitive engine configured to receive primitive commands that include information for processing vertex data using user-developed programs or subroutines. The programmable primitive engine also is configured to transmit program commands that include program pointers and data pointers. In addition, the programmable geometry engine includes a processing engine configured to receive the program commands. The processing engine is further configured to retrieve the user-developed programs or subroutines using the program pointers and to retrieve vertex data using the data pointers. Also, the processing engine is configured to process the vertex data based on instructions included in the user-developed programs or subroutines to produce processed vertex data and to transmit results to the programmable primitive engine.

U.S. Patent No. 6,894,689 to Greene et al. discloses a system, method and computer program product for avoiding reading z-values in a graphics pipeline. Initially, near z-values are stored which are each representative of a near z-value on an object in a region. Such region is defined by a tile and a coverage mask therein. Thereafter, the stored near z-values are compared with far z-values computed for other objects in the region. Such comparison indicates whether an object is visible in the region. Based on the comparison, z-values previously stored for image samples in the region are conditionally read from memory.

U.S. Patent No. 6,894,687 to Kilgard et al. discloses a system, method and article of manufacture for aliasing vertex attributes during vertex processing. Initially, a plurality of identifiers are each mapped to one of a plurality of parameters associated with vertex data. Thereafter, the vertex data is processed by calling the parameters utilizing a vertex program capable of referencing the parameters using the identifiers.

U.S. Patent No. 6,876,362 to Newhall, Jr. et al. discloses an invention for rendering using an omnidirectional light. A shadow cube texture map having six cube faces centered by a light source is generated. Each cube face comprises a shadow texture having depth data from a perspective of the light source. In addition, each cube face is associated with an axis of a three-dimensional coordinate system. For each object fragment rendered from the camera's perspective a light-to-surface vector is defined from the light source to the object fragment, and particular texels within particular cube faces are selected based on the light-to-surface vector. The texel values are tested against a depth value computed from the light to surface vector. The object fragment is textured as in light or shadow according to the outcome of the test.

U.S. Patent No. 6,870,540 to Lindholm et al. discloses a system, method and computer program product for programmable pixel processing in a computer graphics pipeline. Initially, pixel data is received from a source buffer. Thereafter, programmable operations are performed on the pixel data in order to generate output. The operations are programmable in that a user may utilize instructions from a predetermined instruction set for generating the same. Such output is stored in a register.

U.S. Patent No. 6,864,893 to Zatz discloses a method and apparatus for generating depth values in a programmable graphics system. Depth values are calculated under control of a pixel program using a variety of sources as inputs to programmable computation units (PCUs) in the programmable graphics systems. The PCUs are used to compute traditional interpolated depth values and modified depth values. Th PCUs are also used to compute arbitrary depth values which, unlike traditional interpolated depth values and modified depth values, are not dependent on the coordinates of the geometry primitive with which the arbitrary depth values are associated. Several sources are available as inputs to the PCUs. Clipping with optional clamping is performed using either interpolated depth values or calculated depth values, where calculated depth values are arbitrary depth values or modified depth values. Final depth values, used for depth testing, are selected from interpolated depth values and arbitrary depth values after clipping is performed.

U.S. Patent No. 6,856,320 to Rubinstein et al. discloses a memory system and methods of operating the same that drastically increase the efficiency in memory use and allocation in graphics systems. In a graphics system using a tiled architecture, instead of pre-allocating a fixed amount of memory for each tile, the invention dynamically allocates varying amounts of memory per tile depending on the demand. In one embodiment all or a portion of the available memory is divided into smaller pages that are preferably equal in size. Memory allocation is done by page based on the amount of memory required for a given tile.

U.S. Patent No. 6,831,652 to Orr discloses a specific implementation of the present invention, wherein the control portion of a graphics processor receives a command having both a data portion and a data duration portion. When the data duration portion indicates the data is transient data for short-term use, the control portion stores the data associated with the data portion at the first memory partition. When the data duration portion indicates the data is persistent data for long-term use, the control portion stores the data associated with the data portion at a second memory partition. In a multiple processor system, transient data may be stored only in a memory partition associated with a first processor, while persistent data may be stored in multiple memory partitions, one for each graphics processor.

U.S. Patent No. 6,828,987 to Swan discloses a method and apparatus for using multiple scalers to scale video and graphics data. A video data is capable of accessing data formats not accessible by a graphics engine, pipeline, or scaler. The video data can be accessed by the video scaler and subsequently converted to a data format accessible by the graphics engine, pipeline or scaler. The converted video data can be routed to memory. Once stored in memory, the data can be accessed for further processing, such as additional scaling or graphics processing. In addition, the converted video data can be output directly to a processor, such as a graphics processor for additional processing.

U.S. Patent No. 6,828,980 to Moreton et al. discloses a system, method and computer program product for computer graphics processing. Initially, a height parameter is determined. Thereafter, a depth-direction component of the height parameter is calculated. A depth-value of a pixel is then modified utilizing the computed depth-direction component of the height parameter.

U.S. Patent No. 6,825,843 to Allen et al. discloses a method and apparatus for executing loop and branch program instructions in a programmable graphics shader. The programmable graphics shader converts a sequence of instructions comprising a portion of a shader program and selects a first set of fragments to be processed. Subsequent sequences of instructions are converted until all of the instructions comprising the shader program have been executed on the first set of fragments. Each remaining set of fragments is processed by the shader program until all of the fragments are processed in the same manner. Furthermore, the instructions can contain one or more loop or branch program instructions that are conditionally executed. Additionally, when instructions within a loop as defined by a loop instruction are being executed a current loop count is pipelined through the programmable graphics shader and used as an index to access graphics memory.

U.S. Patent No. 6,812,927 to Cutler et al. discloses a system and method for reducing the number of depth clear operations in a hardware graphics pipeline. Initially, a frame count is stored into a frame buffer associated with the hardware graphics pipeline. The stored frame count is associated with a pixel. A depth clear operation is then performed based at least in part on the frame count utilizing the hardware graphics pipeline.

U.S. Patent No. 6,797,998 to Dewey et al. discloses a multi-configuration interface device for coupling different types of GPUs (graphics processor units) to a PCB (printed circuit board). The interface device comprises a GPU interface for a connection to the GPU and a PCB interface for a connection to the PCB. The GPU interface is implemented using a customizable attachment footprint for effectuating a connection to differing GPU types while maintaining the PCB interface for the connection to the PCB. The ball array for different GPUs can be configured to respectively support them. The interface device maintains a consistent PCB interface. Thus, as GPU characteristics change and evolve, or as different GPU versions are implemented, a consistent connection can be maintained for the PCB.

U.S. Patent No. 6,779,069 to Triechler et al. discloses an apparatus for a baseband-media interface. More particularly, in an embodiment, a baseband processor, a medium access controller and a baseband-media interface are provided with a input/output controller as an integrated circuit. In another instance, a baseband processor, a medium access controller and a

baseband-media interface are provided on a printed circuit board and coupled to an input/output controller via a bus. The printed circuit board may be a system board or a peripheral card.

U.S. Patent No. 6,778,189 to Kilgard discloses a system, method and computer program product for two-sided stencil testing during graphics processing. Initially, primitives are received to be processed in a graphics processing pipeline. In use, it is then determined whether the graphics processing pipeline is operating with same-sided stencil testing enabled. If same-sided stencil testing is not enabled, the primitives are passed without same-sided stencil testing and two-sided stencil testing. If, on other hand, same-sided stencil testing is enabled, it is determined whether the graphics processing pipeline is operating with two-sided stencil testing enabled. If the two-sided stencil testing is enabled and the same-sided stencil testing is enabled, two-sided stencil testing is performed on the primitives. If, on the other hand, the two-sided stencil testing is disabled and the same-sided stencil testing is enabled, same-sided stencil testing is performed on the primitives.

U.S. Patent No. 6,778,181 to Kilgariff et al. discloses a graphics processing system. The graphics processing system includes a front end module for receiving pixel data. A setup unit is coupled to the front end module and generates parameter coefficients. A raster unit is coupled to the setup unit and generates stepping information. A virtual texturing array engine textures and colors the pixel data based on the parameter coefficients and stepping information. Also provided is a pixel engine adapted for processing the textured and colored pixel data received from the virtual texturing array engine.

U.S. Patent No. 6,778,176 to Lindholm et al. discloses a method, apparatus and article of manufacture for sequencing graphics processing in a transform or lighting operation. A plurality of mode bits are first received which are indicative of the status of a plurality of modes of process operations. A plurality of addresses are then identified in memory based on the mode bits. Such addresses are then accessed in the memory for retrieving code segments which each are adapted to carry out the process operations in accordance with the status of the modes. The code segments are subsequently executed within a transform or lighting module for processing vertex data.

U.S. Patent No. 6,774,895 to Papakipos et al. discloses a system, method and computer program product for depth clamping in a hardware graphics pipeline. It is then determined as to whether a hardware graphics pipeline is operating in a depth clamping mode. If the hardware graphics pipeline is operating in the depth clamping mode, the depth value is clamped within a predetermined range utilizing the hardware graphics pipeline.

U.S. Patent No. 6,744,433 to Bastos et al. discloses a system and method for using information from at least one depth layer and for collecting information about at least one additional depth layer utilizing a graphics pipeline. Initially, constraining depth layers are provided which, in turn, define a plurality of depth constraints. Next, a plurality of tests is performed involving the constraining depth layers for collecting information about at least one additional depth layer. The information relating to the at least one depth layer may then be used to improve processing in the graphics pipeline. By the foregoing multiple tests, information relating to a plurality of depth layers may be collected during each of a plurality of rendering passes. Initially, information relating to the constraining depth layers and associated depth

constraints is provided in the aforementioned manner. Thereafter, information relating to at least one additional depth layer is collected during additional rendering passes using multiple tests on each rendering pass. Once collected, such information relating to the constraining depth layers and the information relating to the at least one additional depth layer may be used to further improve processing in the graphics pipeline.

U.S. Patent No. 6,734,874 to Lindholm et al. discloses a method, apparatus and article of manufacture for handling both scalar and vector components during graphics processing. To accomplish this, vertex data is received in the form of vectors after which vector operations are performed on the vector vertex data. Next, scalar operations may be executed on an output of the vector operations, thereby rendering vertex data in the form of scalars. Such scalar vertex data may then be converted to vector vertex data for performing vector operations thereon.

U.S. Patent No. 6,734,861 to Van Dyke et al. discloses a system, method and article of manufacture for providing an interlock module in a graphics pipeline. initially, first information is received indicative of a first set of pixels that overlap a primitive. Such first set of pixels are currently being processed in the graphics pipeline. Also received is second information indicative of a second set of pixels that overlap the primitive. The second set of pixels are ready for being inputted in the graphics pipeline for processing. Thereafter, the first information and the second information are evaluated, and the second set of pixels is conditionally processed based on the evaluation.

U.S. Patent No. 6,731,298 to Moreton et al. discloses a system, method and article of manufacture for computer graphics processing. First, pixel data is received including a depth-value. Thereafter, the depth-value is modified based on a depth-component of an algorithm. An operation is subsequently performed on the pixel data taking into account the modified depth-value.

U.S. Patent No. 6,728,820 to Brian et al. discloses a system for providing video, the system having a system bus, which in one embodiment is an Advanced Graphics Port (AGP) bus. The system bus is connected to a data bridge, which is connected to a second and third AGP bus. Each of the AGP busses are connected to graphics processors. The bridge routes data requests from one graphics processor to the second graphics processor without accessing the system AGP bus based upon a memory mapping information stored in a routing table or a register set. In another aspect of the present invention, the bridge responds to initialization requests using attributes that may vary depending on the specific mode of operation. Another aspect of the present invention allows for conversion between various AGP protocol portions.

U.S. Patent No. 6,725,457 to Priem et al. discloses a process of coordinating access to a shared resource by a plurality of execution units. Channel control units are used to coordinate access to a shared resource. Each channel control unit reads semaphore values of a semaphore storage unit. In response to synchronization commands and semaphore values, the channel control unit manages the flow of execution instructions to the execution units in order to manage access to the shared resource.

U.S. Patent No. 6,724,394 to Zatz et al. discloses a system and associated method for processing pixel data in a graphics pipeline. Included is a triangle module coupled to a rasterizer

for calculating a plurality of equations using pixel data received from the rasterizer. Also provided is a shader core module coupled to the rasterizer for receiving the pixel data therefrom. The shader core module is further coupled to the triangle module for receiving the equations therefrom. The shader core module functions to execute floating point calculations and generating texture coordinates using the pixel data. Coupled to the shader core module is a texture module. The texture module is capable of looking up texture values using the texture coordinates. Associated therewith is a shader back end module coupled to the texture module and the triangle module. The shader back end module is capable of converting the texture values to an appropriate floating point representation and generating color values using the equations. Still yet, a combiner module is coupled to the shader core module and the shader back end module. Such combiner module combines the color values and the texture values.

U.S. Patent No. 6,704,025 to Bastos et al. discloses a system and method for improved shadow mapping in a graphics pipeline. Raw depth values are initially collected from two depth layers in a scene to be rendered. Shadow-map depth values are then calculated utilizing the raw depth values. The scene is then shadow mapped utilizing the shadow-map depth values in order to improve the appearance of shadows in a rendered scene. The various steps are carried out by a hardware-implemented graphics pipeline, which may include texturing or shadowing mapping hardware.

U.S. Patent No. 6,700,583 to Fowler et al. discloses a configurable buffer has two storage areas. Depending upon a state of a buffer control signal, the two storage areas are configured to buffer a single stream of data together or to buffer two streams of data separately. In an exemplary video graphics processing application, one stream of data includes pass-through values of fragments being rendered (e.g. color, location, and/or depth values) and the other stream of data includes corresponding displaced (or otherwise perturbed) texture coordinate pairs. Such a buffer may be used to reduce the amount of buffer storage needed to support both single-pass and multipass operations in a pixel pipeline.

U.S. Patent No. 6,691,180 to Priem et al. discloses a direct memory access (DMA) circuit which is physically positioned with an input/output device, the DMA circuit storing a first reference value pointing to a data structure which describes a buffer portion of system memory in which data is stored for transfer to the I/O device, a value determining a position within the buffer portion of system memory beginning at which a next sequence of data is to be placed, and a value determining a position within the buffer portion of system memory from which a next sequences of data is to be copied to the I/O device, the DMA circuit including circuitry for reading data from the buffer portion of system memory beginning at the position from which a next sequences of data is to be copied and for writing the data read to the I/O device.

U.S. Patent No. 6,690,372 to Donovan et al. discloses a system, method and article of manufacture for shadow mapping while rendering a primitive in a graphics pipeline. Initially, an offset operation is performed in order to generate a depth value while rendering a primitive. Further, a value of a slope associated with an edge of the primitive is identified. Thereafter, the depth value is conditionally clamped based on the value of the slope.

U.S. Patent No. 6,677,953 to Twardowski et al. discloses a system and method for a dedicated hardware-implemented viewport operation in a graphics pipeline. Included is a

transform/lighting module for transforming and lighting vertex data. Also provided is viewport hardware coupled to the transform/lighting module for performing a viewport operation on the vertex data. A rasterizer is coupled to the viewport hardware for rendering the vertex data.

U.S. Patent No. 6,670,958 to Aleksic et al. discloses a system for providing video which includes a system bus, which in one embodiment is an Advanced Graphics Port (AGP) busy. The system bus is connected to a data bridge, which is connected to a second and third AGP bus. Each of the AGP busses are connected to graphics processors. The bridge routes data requests from one graphics processor to the second graphics processor without accessing the system AGP bus based upon a memory mapping information stored in a routing table or a register set. In another aspect of the present invention, the bridge responds to initialization requests using attributes that may vary depending on the specific mode of operation. Another aspect of the present invention allows for conversion between various AGP protocol portions.

U.S. Patent No. 6,664,963 to Zatz discloses a system, method and computer program product for performing shader calculations in a graphics pipeline. Initially, a shading calculation is performed in order to generate output. Thereafter, an additional shading calculation is carried out. Such additional shading calculation includes converting the output of the shading calculation into a floating point format. Further, a dot product is calculated utilizing the converted output and texture coordinates. The dot product is then clamped. Next, the clamped dot product is stored in a plurality of color components.

U.S. Patent No. 6,664,960 to Goel et al. discloses an apparatus for processing a non-planar graphics primitive employs an associated method of operation and includes a controller, at least one computation engine, memory and at least one lookup table. Responsive to operation codes issued by the controller, the computation engine(s) determines a group of control points based on the position coordinates and normal vectors of the non-planar primitive vertices. The computation engine(s) then determines position coordinates of supplemental vertices defining multiple planar tessellated primitives based on the control points and stored weighting factors that provide a cubic relation between the control points and the position coordinates of the supplemental vertices. A first memory stores at least the control points and at least one lookup table stores the cubic weighting factors. A second memory stores the position coordinates of the non-planar primitive vertices and the supplemental vertices of the planar primitives generated through tessellation for further graphics processing.

U.S. Patent No. 6,662,257 to Caruk et al. discloses a system for providing video, wherein the system has a system bus, which in one embodiment is an Advanced Graphics Port (AGP) bus. The system bus is connected to a data bridge, which is connected to a second and third AGP bus. Each of the AGP busses are connected to graphics processors. The bridge routes data requests from one graphics processor to the second graphics processor without accessing the system AGP bus based upon a memory mapping information stored in a routing table or a register set. In another aspect of the present invention, the bridge responds to initialization requests using attributes that may vary depending on the specific mode of operation. Another aspect of the present invention allows for conversion between various AGP protocol portions.

U.S. Patent No. 6,657,635 to Hutchins et al. discloses methods and systems for optimizing graphics data processing employ various binning flush algorithms to optimize the

utilization of binning memory in a graphics system. Binning flush algorithms provide for processing all geometry and commands binned up to the point the binning memory becomes unavailable, and storing and restoring all necessary intermediate data generated during the partial tile rendering.

U.S. Patent No. 6,650,331 to Lindholm et al. discloses a graphics pipeline system with an integrated scissor operation. First provided is a transform module adapted for being coupled to a buffer to receive graphics data therefrom. Such transform module is positioned on a single semiconductor platform for transforming the graphics data from a first space to a second space. Associated therewith is a lighting module coupled to the transform module and positioned on the same single semiconductor platform as the transform module for performing lighting operations on the graphics data received from the transform module. A scissor operation is performed on the same single semiconductor platform as the transform module and the lighting module.

U.S. Patent No. 6,650,330 to Lindholm et al. discloses a method, apparatus and article of manufacture for sequencing graphics processing in a transform or lighting operation. A plurality of mode bits are first received which are indicative of the status of a plurality of modes of process operations. A plurality of addresses are then identified in memory based on the mode bits. Such addresses are then accessed in the memory for retrieving code segments which each are adapted to carry out the process operations in accordance with the status of the modes. The code segments are subsequently executed within a transform or lighting module for processing vertex data.

U.S. Patent No. 6,646,639 to Greene et al. discloses a system, method and computer program product for avoiding reading z-values in a graphics pipeline. Initially, near z-values are stored which are each representative of a near z-value on an object in a region. Such region is defined by a tile and a coverage mask therein. Thereafter, the stored near z-values are compared with far z-values computed for other objects in the region. Such comparison indicates whether an object is visible in the region. Based on the comparison, z-values previously stored for image samples in the region are conditionally read from memory.

U.S. Patent No. 6,636,215 to Greene discloses a z-buffer-system having a host processor and graphics hardware that performs hierarchical z-buffering. The z-buffer system renders three-dimensional scenes having geometric primitives that are organized in bounding boxes or rooms-with-portals. As an image is being generated, some but not all z-pyramid values are written from the graphics system into memory that can be quickly accessed by the host processor. This enables the host processor to perform visibility tests that cull occluded bounding boxes or portals, thereby accelerating rendering by reducing the number of primitives that need to be sent to graphics hardware and processed.

U.S. Patent No. 6,636,212 to Zhu discloses a display which is partitioned into a plurality of cells. Each of the plurality of cells includes a depth interval and a coverage for each of a plurality of surfaces having coverage within the cell. A depth interval and a coverage for a group of pixels having coverage within one or more of the cells is received. Visibility of the group of pixels, for each of the one or more cells, is determined based on comparisons of the depth interval of the group of pixels with the depth intervals of the plurality of surfaces.

U.S. Patent No. 6,633,296 to Laksono et al. discloses a system for providing video, the system having a system bus, which in one embodiment is an Advanced Graphics Port (AGP) bus. The system bus is connected to a data bridge, which is connected to a second and third AGP bus. Each of the AGP busses are connected to graphics processors. The bridge routes data requests from one graphics processor to the second graphics processor without accessing the system AGP bus based upon a memory mapping information stored in a routing table or a register set. In another aspect of the present invention, the bridge responds to initialization requests using attributes that may vary depending on the specific mode of operation. Another aspect of the present invention allows for conversion between various AGP protocol portions.

U.S. Patent No. 6,593,923 to Donovan et al. discloses a system, method and article of manufacture for shadow mapping while rendering a primitive in a graphics pipeline. Initially, an offset operation is performed in order to generate a depth value while rendering a primitive. Further, a value of a slope associated with an edge of the primitive is identified. Thereafter, the depth value is conditionally clamped based on the value of the slope.

U.S. Patent No. 6,577,320 to Kirk discloses a method for processing multiple types of pixel component representations. The method first includes identifying a plurality of texels in a texture pattern grid that correspond to a pixel. Thereafter, information components of the pixel, i.e. R, G, B, and .alpha. are multiplied if the information components of the pixel are in a postmultiplied representation. Further, a colorkeyed replacement operation is carried out if the information components of the pixel are in a colorkeyed representation and at least one of the texels substantially matches a colorkey. Next, a position is interpolated on the texture pattern grid between the texels that corresponds to the pixel. Finally, the information components of the pixel are filtered.

U.S. Patent No. 6,577,309 to Lindholm et al. discloses a graphics pipeline system with a transform module positioned on a single semiconductor platform for transforming graphics data. Also included is a lighting module positioned on the same single semiconductor platform as the transform module for lighting the graphics data. In use, various operations may be performed utilizing the single semiconductor platform such as rendering, fog operations, blending, coloring operations, etc.

U.S. Patent No. 6,542,971 to Reed discloses a buffering system attached to a memory for holding write-once, read-once data that is accessed by one or more peripheral devices. Data that is otherwise destined to be written to main memory is written, instead, into a storage buffer. The buffer is written using an address contained in a write pointer that is updated according to a predetermined pattern after the write operation. After updating the write pointer, if the address equals the read pointer, some or all of the buffer is flushed to the memory. Data is read from the buffer using an address contained in a read pointer that is updated according to the same predetermined pattern after the read operation. Any deviation from the pattern in either writing or reading the buffer causes the some or all of the buffer to be flushed to main memory and the read pointer to be updated accordingly.

U.S. Patent No. 6,535,209 to Abdalla et al. discloses a computer graphics system which splits vertex data into first and second streams and stores the streams in separate regions of memory. In a specific embodiment, the first stream includes positional data and the second

stream includes non-positional color and texture data. a visibility subsystem uses only the first stream to perform visibility processing, thus reducing bandwidth requirement. The rendering system processes data from subsets, identified by the visibility subsystem, of both streams required to render the visible part of a scene.

U.S. Patent No. 6,532,525 to Aleksic et al. discloses a method and apparatus for processing data access requests from a requesting device, such as a graphics processor device. Data access commands are provided at a first rate, for example 200M commands per second, to a memory bridge. In response to receiving the access requests the memory bridge will provide its own access requests to a plurality of memories at approximately the first rate. In response to the memory bridge requests, the plurality of memories will access a plurality of data at a second data rate. When the data access between the memory bridge and the memories is a read request, data is returned to the requesting device at a third data rate which is greater than the first data rate by approximately four-times or more. Noise and power reduction techniques can be used on the data bus between the accessing device and the data bridge.

U.S. Patent No. 6,532,013 to Papakipos et al. discloses a system, method and article of manufacture for interweaving shading calculations and texture retrieval operations during texture sampling in a graphics pipeline. First, a shading calculation is performed in order to generate output. Next, texture information is retrieved, and another shading calculation is performed using the texture information in order to generate additional output. Texture information may be retrieved and shading calculations may then be repeated as desired. Thereafter, the generated output may be combined. As such, the repeated texture information retrieval and shading calculations may be carried out in an iterative, programmable manner.

U.S. Patent No. 6,502,173 to Aleksic et al. discloses a method and apparatus for processing data access requests from a requesting device, such as a graphics processor device. Data access commands are provided at a first rate, for example 200M command per second, to a memory bridge. In response to receiving the access requests the memory bridge will provide its own access requests to a plurality of memories at approximately the first rate. In response to the memory bridge requests, the plurality of memories will access a plurality of data a second data rate. When the data access between the memory bridge and the memories is a read request, data is returned to the requesting device at a third data rate which is greater than the first data rate by approximately four times or more. Noise and power reduction techniques can be used on the data bus between the accessing device and the data bridge.

U.S. Patent No. 6,496,404 to Fiedler et al. discloses a memory system comprising a circuit board and at least two memory devices mounted on the circuit board. Each of the at least two memory devices includes a plurality of pins for receiving and providing signals. At least a first portion of the pins of one of the at least two memory devices are coupled to at least a second portion of the pins of the other at least two memory devices such that a pair of the first portion coupled to a pin of the second portion forms a coupled load. The coupled load then appears as one load. Accordingly, in a system in accordance with the present invention, at least two memory devices are provided on a circuit board. Each of the at least two memory devices includes a plurality of pins. At least a portion of the pins of one of the two memory devices is in close proximity to and coupled to the at least a portion of the pins of the other of the at least two memory devices such that a pin and one memory device is coupled to a pin on the other memory

device to form a coupled load. The coupled load then appears as one load. This is accomplished in a preferred embodiment by allowing the pins which are on opposite sides (front and back) of a printed circuit board to be represented as one load and then remapping one of the oppositely disposed pins to have the same functionality as the other oppositely disposed pin.

U.S. Patent No. 6,492,987 to Morein discloses a method and apparatus for processing object elements as they are being rendered, which begins by comparing geometric properties (e.g., the Z value) of at least one object element with a representative geometric property for a group of pixels. The group of pixels is determined based on physical coordinates (i.e., the coordinates of a display) that are inclusive of the physical coordinates of the object element. When the comparison of the geometric properties of the object element compares unfavorably with the representative geometric property for the group of pixels, the object-element is rendered and a Z buffer is updated with new Z buffer information. At a given point during the rendering of the object-element, a new representative geometric property is determined based on the updated Z value. Once the new representative geometric property is determined, the representative geometric property is updated with that new value.

U.S. Patent No. 6,477,687 to Thomas discloses Macrocells, e.g., Random Access Memory ("RAM"), which are arranged in columns and disposed in a core of an integrated circuit (IC) chip. The macrocells can abut each other within the columns or can be separated from each other by standard cells which are disposed to fill gaps between the macrocells within the columns. Power/ground rails are disposed vertically along the sides of the columns. The power/ground rails run the full height of the core and couple to a power/ground ring disposed along the perimeter of the core. The power/ground rails also couple to the macrocells and the standard cells and provide power to those cells. The columns can form right angles with horizontal standard cell rows, thus enabling the standard cells to couple easily to the vertically disposed power/ground rails.

U.S. Patent No. 6,473,089 to Wei et al. discloses a method and apparatus for parallel processing of pixel information within a video graphics circuit that is accomplished when the video graphics circuit includes a set-up engine, an edgewalker circuit, a span processing circuit, and a plurality of pixel processing circuits. In such an embodiment, the set-up engine receives vertex information and produces object-element information therefrom. The object-element information is provided to the edgewalker circuit, which in turn produces span definition information. The span definition information identifies the starting pixel of a span and the starting pixel parameters. The span information is received by the processing circuit and converted into a plurality of pixel parameters. The plurality of pixel parameters are provided to the plurality of pixel processing circuits wherein each of the plurality of pixel processing circuits processes corresponding pixel parameters to produce pixel information in accordance with the information provided by the processing circuit.

U.S. Patent No. 6,473,086 to Morein et al. discloses a method and apparatus for graphics processing that utilizes multiple graphics processors in parallel. A primary graphics processor is operably coupled to a primary memory that includes a primary color buffer and a primary Z buffer. The primary processor processes a first portion of the image data for a frame, where processing the first portion stores color data in the primary color buffer and Z data in the primary Z buffer. A secondary processor is operably coupled to a secondary memory that includes a

secondary color buffer and a secondary Z buffer. The secondary processor processes a second portion of the image data for the frame. The processing of the second portion of the image data results in color data being stored in the secondary color buffer and Z data being stored in the secondary Z buffer. The display signal that results in the image data for the frame being displayed is generated by a display driver included in the primary processor. In one embodiment, the display driver retrieves all of the color data used to generate the display signal from the primary color buffer. As such, the secondary processor transfers the color data for the second portion of the frame from the secondary color buffer to the primary color buffer to facilitate generation of the display signal. This data transference preferably occurs utilizing direct memory access (DMA) transfers that may be initiated during the vertical blanking interval portion of the display signal.

U.S. Patent No. 6,462,737 to Lindholm et al. discloses a graphics pipeline system with an integrated clipping operation. First included is a transform module adapted for being coupled to a buffer to receive graphics data therefrom. Such transform module is positioned on a single semiconductor platform for transforming the graphics data from a first space to a second space. Also provided is a lighting module coupled to the transform module and positioned on the same single semiconductor platform as the transform module. The lighting module is adapted for performing lighting operations on the graphics data received from the transform module. A range clamp inversion function and a clipping operation are performed on the same single semiconductor platform as the transform module and the lighting module.

U.S. Patent No. 6,442,656 to Alasti et al. discloses a method and apparatus for interfacing memory with a bus in a computer system which includes processing that begins by receiving a transaction from the bus. The transaction may be a read transaction and/or a write transaction. Upon receiving the transaction, the process continues by validating the received transaction and, when valid, acknowledges its receipt. The processing then continues by storing the physical address, which was included in the received transaction, and the corresponding command in an address/control buffer. The processing continues by retrieving the physical address from the address/control buffer when the transaction is to be processed. The determination of when the transaction is to be processed is based on an ordering within the address/control buffer. The processing then continues by performing the transaction utilizing a first or second memory path based on the physical address, such that a first or second memory is accessed.

U.S. Patent No. 6,415,345 to Wu et al. discloses a bus interface control system and method which includes an on-demand bus master interface for independently requesting multistream data from host memory without interrupting processing of the host processor between independent requests for data packets. A plurality of digital signal processors share the host bus and utilize flexible data speed transfer depending upon demand of real time data that must be transferred from host memory. The master interface control system includes an packet by packet arbitor to facilitate maximum throughput of data on-demand by the plurality of processing unit.

U.S. Patent No. 6,352,479 to Sparks, II discloses a multiplayer game system over the WWW using a plurality of game servers dynamically linked to and controlled by a WWW server. The WWW server dynamically links game players who log on to a web site hosted by the WWW server as a function of game playing statistics for each game player which are stored in

the WWW server. The game servers generate the game player statistics for each player during and/or after game play and upload the game player statistics to the WWW server. The WWW server matches game players to appropriate games currently being played on the game servers based on the skill level required by the game and the corresponding skill levels of other current players of that game as represented by the game player statistics stored by the WWW server and dynamically generates links for the game player to the appropriate games. The user can then select which game to play by choosing one of the dynamically generated links.

U.S. Patent No. 6,333,744 to Kirk et al. discloses a graphics pipeline including a rasterizing stage producing diffuse color values; a plurality of texture stages producing texture values defining a particular texture; a combiner stage for combining four of a plurality of selectable input values including diffuse color values, texture values furnished by a plurality of texture stages, and proportions for combination of the selectable input values; the combiner stage being capable of providing a result equivalent to a sum of products of any two sets of input values, and a product of two input values.

U.S. Patent No. 6,337,686 to Wong et al. discloses a method and apparatus for processing line anti-aliasing which begins by walking a mathematical line based on the Bresenham technique. While walking the mathematical line at each pixel along the mathematical line- pixel coverage area is determined for each pixel of a set of pixels, where the set of pixels traverse a minor direction of the mathematical line. Note that for the mathematical line, the minor direction is the X direction when .DELTA.Y is greater than .DELTA.X and is in the Y direction when .DELTA.X is greater than .DELTA.Y. Once the coverage pixel coverage area of each pixel in the set of pixels has been determined, the intensity for each pixel in the set of pixels is determined. The intensity corresponds to the particular RGB value being generated for subsequent display.

U.S. Patent No. 6,288,418 to Reed et al. discloses an integrated circuit including a plurality of connectors for communicating with circuitry within the integrated circuit, a plurality of input/output pads for connecting to external circuitry, a plurality of multiplexors joined to the connectors and the input/output pads, means for providing an external control signal for each multiplexor for joining the plurality of connectors for communicating with circuitry within the integrated circuit to correct input/output pads for connecting to external circuitry for operating the integrated circuit.

U.S. Patent No. 6,201,545 to Wong et al. discloses a method and apparatus for providing video graphics processing that includes anti-aliasing begins when a video graphics processor receives vertex parameters of an object-element and walks a first edge of the object element and a second edge of the object element. The video graphics processor walks the first and second edges based on calculations involving a decision corner and an error turn. The calculations indicate which direction to walk on a pixel by pixel basis, i.e., whether the walking should be in the major direction or the minor direction. The calculations also identify pixels that contain fragment pixel information, i.e., the pixels along the edges of the object element. For each fragment pixel, subpixel masks are created for each object element that is present in the fragmented pixel. From the subpixel masks, subpixel sequences are determined, which are used to produce pixel information of the fragment pixel.

U.S. Patent No. 6,188,412 to Morein discloses a method and apparatus performing setup operations in a video graphics system that is accomplished by receiving a stream of primitives and determining if each primitive in the stream has a parameter that exceeds the calculation range of a fast multiplier. If it is determined that the primitive exceeds the capabilities of the fast multiplier, the calculations for the primitive are routed to a setup engine that includes a full-size multiplier, and the calculations of setup parameters are performed by the full-size multiplier. If it is determined that the calculations are within the capabilities of the fast multiplier, the calculations for the primitive are routed to the setup engine that includes one or more fast multipliers, and the setup calculations are performed therein. The results of the two setup engines are combined to form a resultant stream that includes the parameter calculations required from display setup operations.

U.S. Patent No. 6,184,908 to Chan et al. discloses CPU processing requirements for preparing and transferring data to a graphics processor, wherein a graphics command processor is provided that supports application-level commands and references to the data associated with these commands. The graphics command processor parses the application command and data reference parameters, and subsequently fetches the appropriate graphics data from memory directly, without requiring additional CPU resources. To optimize performance, the graphics command processor fetches the data in parallel with the parsing and processing of the application commands from the CPU. The graphics command processor also includes a processing unit that converts the data from the format used by the application program to the format used for rendering. The graphics command processor creates the commands and data sequences used by a graphics engine to render each object of the image. Because the graphics command processor is closely coupled with the graphics engine, a number of efficiency can be gained, particularly with regard to the transfer of related data items. The processing of the primitive graphic command and data sequences by the graphics engine is asynchronous with the receipt of subsequent commands from the CPU and the fetching of subsequent data associated with the commands from the memory. In this manner, the latency associated with the conventional sequential processing of graphics data is minimized.

U.S. Patent No. 6,181,352 to Kirk et al. discloses a graphics accelerator pipeline including a combiner stage capable of producing output values during each clock interval of the pipeline which map a plurality of textures to a single pixel or an individual texture to two pixels.

U.S. Patent No. 6,169,553 to Fuller et al. discloses a method and apparatus for rendering shadows on a three-dimensional scene which is accomplished by generating a light source texture map of the given three-dimensional scene. The light source texture map is generated based on a viewing perspective being proximal to a light source projection. Once the light source texture map has been generated, an object element of the three-dimensional scene is rendered based on the light source texture map and image information. The image information includes vertex components corresponding to object elements of the three-dimensional scene.

U.S. Patent No. 5,909,595 to Rosenthal et al. discloses a method of controlling the routing of input/output operations which includes providing a series of commands expressing connections between sources of data, processing elements, and destinations for data to carry out an input/output operation; compiling a data structure for the input/output operation from the series of commands, the data structure including context defining connections between each of

the sources of data, processing elements, and destinations for data; and using the data structure to set connecting context to make connection expressed between each of the sources of data, processing elements, and destinations for data whenever the input/output operation is to be accomplished.

U.S. Patent No. 5,758,182 to Rosenthal et al. discloses a DMA controller which responds without operating system intervention to virtual addresses provided by application programs, and a memory management unit for providing translations between physical addresses of input/output devices and addresses on a system input/output bus for data transferred by the DMA controller.

U.S. Patent No. 5,754,866 to Priem discloses an apparatus for transferring commands over a system transmission path between first and second components in a digital data system including a first-in first-out circuit having a plurality of stages arranged in the system transmission path, circuitry for generating a first signal to indicate that a component to which a command in the FIFO circuit is directed is unable to handle an operation commanded, and a delay circuit responsive to the first signal for causing the generation of an interrupt request signal after a preselected time.

U.S. Patent No. 5,740,464 to Priem et al. discloses a hardware input/output address translation apparatus adapted for use in a multitasking computer system including hardware responsive to commands from an unprivileged application program addressed to an input/output address for translating the input/output address to a physical address space of an input/output device and transferring the command to the physical address of an input/output device, and additional hardware responsive to commands from an unprivileged application program addressed to an input/output address for selecting from safe translations of input/output addresses to physical address spaces of input/output devices for the hardware for translating the input/output address to a physical address space of an input/output device.

U.S. Patent No. 5,687,357 to Priem discloses an apparatus and a method by which an application program writing a series of commands to a single destination on an input/output bus increments the addresses to which the commands are addressed as the commands are written so that the commands may be transferred utilizing the burst mode of the input/output bus, and the device receiving the data decodes a large number of sequential addresses to the same destination so that the input/output device transfers all of the commands in the sequence of addresses to the single destination.

U.S. Publication No. 2007/0195099 to Diard et al discloses a system for processing video data which includes a host processor, a first media processing device coupled to a first buffer, the first media processing device configured to perform a first processing task on a frame of video data, and a second media processing device coupled to a second buffer, the second media processing device configured to perform a second processing task on the processed frame of video data. The architecture allows the two devices to have asymmetric video processing capabilities. Thus, the first device may advantageously perform a first task, such as decoding, while the second device performs a second task, such as post processing, according to the respective capabilities of each device, thereby increasing processing efficiency relative to prior art systems. Further, one driver may be used for both devices, enabling applications to take

advantage of the system's accelerated processing capabilities without requiring code changes.

U.S. Publication No. 2007/0159488 to Danskin et al. discloses a parallel array architecture for a graphics processor which includes a multithreaded core array including a plurality of processing clusters, each processing cluster including at least one processing core operable to execute a pixel shader program that generates pixel data from coverage data; a rasterizer configured to generate coverage data for each of a plurality of pixels; and pixel distribution logic configured to deliver the coverage data from the rasterizer to one of the processing clusters in the multithreaded core array. The pixel distribution logic selects one of the processing clusters to which the coverage data for a first pixel is delivered based at least in part on a location of the first pixel within an image area. The processing clusters can be mapped directly to the frame buffers partitions without a crossbar so that pixel data is delivered directly from the processing cluster to the appropriate frame buffer partitions. Alternatively, a crossbar coupled to each of the processing clusters is configured to deliver pixel data from the processing clusters to a frame buffer having a plurality of partitions. The crossbar is configured such that pixel data generated by any one of the processing clusters is deliverable to any one of the frame buffer partitions.

U.S. Publication No. 2006/0282604 to Temkine et al. discloses methods and apparatus for providing multiple graphics processing capacity, while utilizing unused integrated graphics processing circuitry on a bridge circuit along with an external or discrete graphics processing unit. In particular, a bridge circuit includes an integrated graphics processing circuit configured to process graphics jobs. The bridge circuit also includes an interface operable according to interface with a discrete graphics processing circuit. A controller is included with the bridge circuit and responsive whenever the discrete graphics processing circuit is coupled to the interface to cause the integrated graphics processing circuit to process a task of the graphics job in conjunction with operation of the discrete graphics processing circuit that is operable to process another task of the graphics job. Corresponding methods are also disclosed.

U.S. Publication No. 2006/0271713 to Xie et al. discloses a computing device that allows for a flexible allocation of bandwidth among peripheral devices using a peripheral bus. The computing device includes a peripheral bus and at least two slots. The computing device may be used with a single peripheral card or multiple peripheral cards. In a multi-card configuration the invention allows the bandwidth on the peripheral bus to be shared by all the cards. In a single-card configuration, the computing device allows available bandwidth on the peripheral bus to be used by a single card. The device is particularly useful with PCI express compliant expansion cards, such as graphics adapters.

U.S. Publication No. 2006/0268005 to Hutchins et al. discloses a rasterizer stage configured to implement multiple interpolators for graphics pipeline. The rasterizer stage includes a plurality of simultaneously operable low precision interpolators for computing a first set of pixel parameters for pixels of a geometric primitive and a plurality of simultaneously operable high precision interpolators for computing a second set of pixel parameters for pixels of the geometric primitive. The rasterizer stage also includes an output mechanism coupled to the interpolators for routing computed pixel parameters into a memory array. Parameters may be programmably assigned to the interpolators and the results thereof may be programmably assigned to portions of a pixel packet.

U.S. Publication No. 2006/0248241 to Danilak discloses a universal storage bus adaptor that can interface a host computer's bus to any of multiple types of storage devices. The universal serial bus adaptor provides transport layer functionality in such a way that a separate transport layer does not have to be provided for each type of storage device. Anther embodiment of the present invention includes a file management system (or storage stack) that has a read/write chimney configured to enable a READ/WRITE operation to bypass the exception processing and management functionalities of the file management system. Bypassing these functionalities increases the processing efficiency of READ/WRITE operations.

U.S. Publication No. 2006/0267987 to Litchmanov discloses a computing device which includes first and second graphics adapters. A graphics processor of the first graphics adapter acts as a master graphics processor, while a second graphics adapter acts as a slave. The master graphics processor renders graphics to be displayed on multiple separate displays within memory of the first graphics adapter. Images to be displayed on one of the displays are transferred to memory used by the second graphics adapter. The display interface of the second graphics adapter presents images within the memory of the second graphics adapter on at least one of the multiple displays. In this way, device electronics forming the display interface, as well as ports of the second adapter, acting as a slave, may be utilized. In one embodiment, an application creates a single larger image, rendered within the memory of the first graphics adapter. The larger image is then presented as the first and second smaller images on the multiple displays. In this way, an end user sees the multiple images as a single large image across the multiple displays.

U.S. Publication No. 2006/0225061 to Ludwig et al. discloses a method and apparatus for optimizing register allocation during scheduling and execution of program code in a hardware environment. The program code can be compiled to optimize execution given predetermined hardware constraints. The hardware constraints can include the number of register read and write operations that can be performed in a given processor pass. The optimizer can initially schedule the program using virtual registers and a goal of minimizing the amount of active registers at any time. The optimizer reschedules the program to assign the virtual registers to actual physical registers in a manner that minimizes the number of processor passes used to execute the program.

U.S. Publication No. 2006/0221087 to Diard discloses systems and methods for balancing a load among multiple graphics processors that render different portions of a frame. A display area is partitioned into portions for each of two (or more) graphics processors. The graphics processors render their respective portions of a frame and return feedback data indicating completion of the rendering. Based on the feedback data, an imbalance can be detected between respective loads of two of the graphics processors. In the event that an imbalance exists, the display area is re-partitioned to increase a size of the portion assigned to the less heavily loaded processor and to decrease a size of the portion assigned to the more heavily loaded processor.

U.S. Publication No. 2006/0221086 to Diard discloses systems and methods for balancing a load among multiple graphics processors that render different portions of a frame. A display area is partitioned into portions for each of two (or more) graphics processors. The

graphics processors render their respective portions of a frame and return feedback data indicating completion of the rendering. Based on the feedback data, an imbalance can be detected between respective loads of two of the graphics processors. In the event that an imbalance exists, the display area is re-partitioned to increase a size of the portion assigned to the less heavily loaded processor and to decrease a size of the portion assigned to the more heavily loaded processor.

U.S. Publication No. 2006/0208960 to Glen discloses an image processing circuit, such as a graphics accelerator chip or any other suitable circuit, includes display output control logic that is operative to receive a current frame of information from a frame buffer and is operative to process a current frame, such as by providing gamma correction, image scaling, graphics or video overlaying, or other suitable processing, to produce a processed current display frame and stores the processed current display frame back in the frame buffer. Fixed function or dedicated, display type specific temporal processing logic receives the processed current display frame stored in the frame buffer and also obtains at least one previous processed current display frame from the frame buffer and temporally processes pixels from each of the processed current display frame and the previous processed current display frame to produce a temporally compensated display frame for a specific type of display.

U.S. Publication No. 2006/0202941 to Morein et al. discloses a graphics processing circuit which includes a pixel shader operative to provide pixel color information in response to image data representing a scene to be rendered; a texture circuit, coupled to the pixel shader, operative to determine a luminance value to be applied to a pixel of interest based on the luminance values of the pixels that define a plane including the pixel of interest; and a render back end circuit, coupled to the texture circuit, operative to compute the luminance values from a shadow map that specifies the distance from the light source of the nearest object at a plurality of locations. A pixel illumination method includes receiving color information for a pixel to be rendered, defining a plane containing at least one pixel of interest, the plane including a plurality of planar values; comparing the plurality of planar values to a corresponding set of distance values; determining a luminance value for the at least one pixel of interest; and applying the luminance value to the at least one pixel of interest.

U.S. Publication No. 2006/0123142 to Duncan et al. discloses a method and apparatus for providing peer-to-peer data transfer through an interconnecting fabric. The method and apparatus enable a first device to read and/or write data to/from a local memory of a second device by communicating read and write requests across the interconnectivity fabric. Such data transfer can be performed even when the communication protocol of the interconnectivity fabric does not permit such transfers.

U.S. Publication No. 2006/0120376 to Duncan et al. discloses a method and apparatus for providing peer-to-peer data transfer through an interconnecting fabric. The method and apparatus enable a first device to read and/or write data to/from a local memory of a second device by communicating read and write requests across the interconnectivity fabric. Such data transfer can be performed even when the communication protocol of the interconnectivity fabric does not permit such transfers.

U.S. Publication No. 2006/0119607 to Lindholm et al. discloses a graphics processing

unit can queue a large number of texture requests to balance out the variability of texture requests without the need for a large texture request buffer. A dedicated texture request buffer queues the relatively small texture commands and parameters. Additionally, for each queued texture command, an associated set of texture arguments, which are typically much larger than the texture command, are stored in a general purpose register. The texture unit retrieves texture commands from the texture request buffer and then fetches the associated texture arguments from the appropriate general purpose register. The texture arguments may be stored in the general purpose register designated as the destination of the final texture value computed by the texture unit. Because the destination register must be allocated for the final texture value as texture commands are queued, storing the texture arguments in this register does not consume any additional registers.

U.S. Publication No. 2006/0114260 to Diard discloses a CPU selectively programs one or more graphics devices by writing a control command to the command buffer that designates a subset of graphics devices to execute subsequent commands. Graphics devices not designated by the control command will ignore the subsequent commands until re-enabled by the CPU. The non-designated graphics devices will continue to read from the command buffer to maintain synchronization. Subsequent control commands can designate different subsets of graphics devices to execute further subsequent commands. Graphics devices include graphics processing units and graphics coprocessors. A unique identifier is associated with each of the graphics devices. The control command designates a subset of graphics devices according to their respective unique identifiers. The control command includes a number of bits. Each bit is associated with one of the unique identifiers and designates the inclusion of one of the graphics devices in the first subset of graphics devices.

U.S. Publication No. 2006/0101218 to Reed discloses circuits, methods, and apparatus that adaptively control 1T and 2T timing for a memory controller interface. An embodiment of the present invention provides a first memory interface as well as an additional memory interface, each having a number of address and control lines. The address and control lines of the redundant memory interface may be individually enabled and disabled. If a line in the additional interface is enabled, it and its corresponding line in the first interface drive a reduced load and may operate at the higher 1T data rate. If a line in the additional interface is disabled, then its corresponding line in the first interface drives a higher load and may operate at the slower 2T data rate. In either case, the operating speed of the interface may also be considered in determining whether each line operates with 1T or 2T timing.

U.S. Publication No. 2006/0059494 to Wexler et al. discloses embodiments of methods, apparatuses, devices, and/or systems for load balancing two processors, such as for graphics and/or video processing, for example.

U.S. Publication No. 2006/0055695 to Abdalla et al. discloses a fragment processor which includes a fragment shader distributor, a fragment shader collector, and a plurality of fragment shader pipelines. Each fragment shader pipeline executes a fragment shader program on a segment of fragments. The plurality of fragment shader pipelines operate in parallel, executing the same or different fragment shader programs. The fragment shader distributor receives a stream of fragments from a rasterization unit and dispatches a portion of the stream of fragments to a selected fragment shader pipeline until the capacity of the selected fragment

shader pipeline is reached. The fragment shader distributor then selects another fragment shader pipeline. The capacity of each of the fragment shader pipelines is limited by several different resources. As the fragment shader distributor dispatches fragments, it tracks the remaining available resources of the selected fragment shader pipeline. A fragment shader collector retrieves processed fragments from the plurality of fragment shader pipelines.

U.S. Publication No. 2006/0028478 to Rubinstein et al. discloses a computer system which includes an integrated graphics subsystem and a graphics connector for attaching either an auxiliary graphics subsystem or a loopback card. A first bus connection communicates data from the computer system to the integrated graphics subsystem. With a loopback card in place, data travels from the integrated graphics subsystem back to the computer system via a second bus connection. When the auxiliary graphics subsystem is attached, the integrated graphics subsystem operates in a data forwarding mode. Data is communicated to the integrated graphics subsystem via the first bus connection. The integrated graphics subsystem then forwards data to the auxiliary graphics subsystem. A portion of the second bus connection communicates data from the auxiliary graphics subsystem back to the computer system. The auxiliary graphics subsystem communicates display information back to the integrated graphics subsystem, where it is used to control a display device.

U.S. Publication No. 2006/0005178 to Kilgard et al. discloses a system which optimizes two or more stream processing programs based upon the data exchanged between the stream processing programs. The system alternately processes each stream processing program to identify and remove dead program code, thereby improving execution performance. Dead program code is identified by propagating constants received as inputs from other stream processing programs and by analyzing a first stream processing program and determining the outputs of a second stream processing program that are unused by the first stream processing program. The system may perform multiple iterations of this optimization is previous iterations introduce additional constants used as inputs to a stream processing program. Following optimization of the stream processing programs, the optimized stream processing programs are compiled to a format adapted to be executed by a stream processing system.

U.S. Publication No. 2005/0275760 to Gritz et al. discloses methods, apparatuses, devices, and/or systems for modifying a rasterized surface, such as by trimming, for graphics and/or video processing, for example.

U.S. Publication No. 2005/0243215 to Doswald et al. discloses a film mode detector detects film mode of a series of fields of video by comparing pixels in a field adjacent the current field, with corresponding pixels directly above and directly below the pixels in an adjacent field. The number of pixels in the adjacent in time to the current field having (or not having) a value approximately between values of the pixels above and below in the current field is assessed. Film mode for a current field may be detected by monitoring the assessment from field to field. Alternatively or additionally, the detector may detect film mode by assessing for each current field, whether a relatively large or relatively small number of pixels in the immediately previous field have values outside a specified distance of values of a corresponding pixel in the immediately subsequent field, for at least a portion of the immediately previous and subsequent fields. Again, film mode may be detected by monitoring this second assessment from field to field.

U.S. Publication No. 2005/0265064 to Ku et al. discloses circuits, methods, and apparatus for training a phase shift circuit to provide a phase shift for improved data recovery. A specific embodiment of the present invention provides a variable delay cell. A delay through the variable delay cell is changed while training patterns are received. The presence of errors in the received data pattern is tracked, and from the presence or absence of errors a preferred delay is selected and used for receiving data.

U.S. Publication No. 2005/0259103 to Kilgard et al. discloses a system, method and computer program product for programmable processing of fragment data in a computer hardware graphics pipeline. Initially, fragment data is received in a hardware graphics pipeline. It is then determined whether the hardware graphics pipeline is operating in a programmable mode. If it is determined that the hardware graphics pipeline is operating in the programmable mode, programmable operations are performed on the fragment data in order to generate output. The programmable operations are performed in a manner/sequence specified in a graphics application program interface. If it is determined that the hardware graphics pipeline is not operating in the programmable mode, standard graphics application program interface (API) operations are performed on the fragment data in order to generate output.

U.S. Publication No. 2005/0225558 to Morein et al. discloses a memory architecture for use in a graphics processor including a main memory, a level one (L1) cache and a level two (L2) cache, coupled between the main memory and the L1 cache is disclosed. The L2 cache stores overlapping requests to the main memory before the requested information is stored in the L1 cache. In this manner, overlapping requests for previously stored information is retrieved from the faster L2 cache as opposed to the relatively slower main memory.

U.S. Publication No. 2005/0243096 to Possley et al. discloses a memory controller hub which includes a graphics subsystem adapted to perform graphics operations on data, and interface circuitry adapted selectively to couple the graphics subsystem to a local memory through electrical connectors and to couple the memory controller hub to a graphics controller through the electrical connectors.

U.S. Publication No. 2005/0237329 to Rubinstein et al. discloses a graphics processing subsystem which uses system memory as its graphics memory for rendering and scanout of images. To prevent deadlock of the data bus, the graphics processing subsystem may use an alternate virtual channel of the data bus to access additional data from system memory needed to complete a write operation of a first data. In communicating with the system memory, a data packet including extended byte enable information allows the graphics processing subsystem to write large quantities of data with arbitrary byte masking to system memory. To leverage the high degree of two-dimensional locality of rendered image data, the graphics processing subsystem arranges image data in a tiled format in system memory. A tile translation unit converts image data virtual addresses to corresponding system memory addresses. The graphics processing subsystem reads image data from system memory and converts it into a display signal.

U.S. Publication No. 2005/0237327 to Rubinstein et al. discloses a computer system which includes an integrated graphics subsystem and a graphics connector for attaching either an

auxiliary graphics subsystem or a loopback card. A first bus connection communicates data from the computer system to the integrated graphics subsystem. With a loopback card in place, data travels from the integrated graphics subsystem back to the computer system via a second bus connection. When the auxiliary graphics subsystem is attached, the integrated graphics subsystem operates in a data forwarding mode. Data is communicated to the integrated graphics subsystem via the first bus connection. The integrated graphics subsystem then forwards data to the auxiliary graphics subsystem. A portion of the second bus connection communicates data from the auxiliary graphics subsystem back to the computer system. The auxiliary graphics subsystem communicates display information back to the integrated graphics subsystem, where it is used to control a display device.

U.S. Publication No. 2005/0223124 to Reed discloses circuits, apparatus, and methods for avoiding deadlock conditions in a bus fabric. One exemplary embodiment provides an address decoder for determining whether a received posted request is a peer-to-peer request. If it is, the posted request is sent as a non-posted request. A limit on the number of pending non-posted requests is maintained and not exceed, such that deadlock is avoided. Another exemplary embodiment provides an arbiter that tracks a number of pending posted requests. When the number pending posted requests reaches a predetermined or programmable level, a Block Peer-to-Peer signal is sent to the arbiter's clients, again avoiding deadlock.

U.S. Publication No. 2005/0206646 to Alcorn discloses a distributed resource system which comprises a plurality of compute resource units operable to execute graphics applications and generate graphics data, and a plurality of visualization resource units communicatively coupled to the plurality of compute resource units and operable to render pixel data from the graphics data. A first network couples a network compositor to the plurality of visualization resource units. The network compositor is operable to synchronize the received pixel data from the plurality of visualization resource units and receive the pixel data from the visualization resource units and to composite the synchronized pixel data into at least one image. A plurality of display devices, at least one of which is located remotely from the plurality of compute resource units, are coupled to the network compositor and operable to display the at least one image.

U.S. Publication No. 2005/0195187 to Seiler et al. discloses a method and apparatus for hierarchical Z buffering stenciling includes comparing an input tile Z value range with a hierarchical Z value range and a stencil code. The method and apparatus also updates the hierarchical Z value range and stencil code in response the comparison and determines whether to render a plurality of pixels within the input tile based on the comparison of the input tile Z value range with the hierarchical Z value range and stencil code. In determining whether to render the tile, a stencil test and a hierarchical Z value test is performed. If one of the test fails, the tile is killed as it is determined that the pixels are not visible in the graphical output. If the stencil test passes and the hierarchical Z test passes, the pixels within the tile are rendered, as it is determined that the pixels may be visible.

U.S. Publication No. 2005/0195186 to Mitchell et al. discloses a method and apparatus for object-based visibility culling which includes receiving a plurality of draw packets, such as pixels or vertices. The method and apparatus further includes comparing each of the plurality of draw packets to a bounding volume object, wherein the bounding volume object may be a low

resolution geometric representation of a specific object. Whereupon, for each of the plurality of draw packets, if the draw packet is deemed potentially visible, setting a visibility query identifier and rendering the draw packets having the set visibility query identifier.

U.S. Publication No. 2005/0190190 to Diard et al. discloses a bridge associated with a broadcast aperture which facilitates the transfer of rendering commands and data between a processor and multiple graphics devices. The bridge receives data written by the processor to the broadcast aperture and forwards it to multiple graphics devices, eliminating the need for the processor to perform duplicative(?) write operations. During system initialization, a broadcast aperture is allocated to the bridge in address space based on an aperture size value set using a system configuration utility and stored in system configuration memory. A graphics driver activates the broadcast aperture by sending unicast aperture parameters associated with the multiple graphics devices to the bridge via a bridge driver. Upon activating the broadcast aperture, multiple graphics devices can be operated in parallel to improve rendering performance. Parallel rendering techniques include split-frame, alternate frame, and combined split- and alternate frame rendering.

U.S. Publication No. 2005/0162437 to Morein et al. discloses a method and apparatus for graphics processing using state and shader management which includes at least one state and shader cache coupled to a compiler for compiling a hardware state and shader vector from an abstract state vector. Also included is an abstract state vector register containing the abstract state vector that is provided to the state and shader cache and the compiler. The state and shader cache receives the abstract state vector and determines whether a cache entry for that abstract state vector already exists. If the cache entry exists, the hardware state and shader vector is provided to hardware. If the entry does not exist, the state and shader cache provides a miss signal to the compiler, whereupon the compiler compiles the abstract state vector and generates a hardware state and shader vector. Thereupon the hardware state and shader vector is provided to the hardware.

U.S. Publication No. 2005/0081115 to Cheng et al. discloses a circuit monitors and resets a co-processor. The circuit includes a hang detector module for detecting a hang in co-processor. The circuit also includes a selective processor reset module for resetting the co-processor without resetting a processor in response to detecting a hang in the co-processor.

U.S. Publication No. 2005/0041031 to Diard discloses systems and methods for balancing a load among multiple graphics processors that render different portions of a frame. A display area is partitioned into portions for each of two (or more) graphics processors. The graphics processors render their respective portions of a frame and return feedback data indicating completion of the rendering. Based on the feedback data, an imbalance can be detected between respective loads of two of the graphics processors. In the event that an imbalance exists, the display area is re-partitioned to increase a size of the portion assigned to the less heavily loaded processor and to decrease a size of the portion assigned to the more heavily loaded processor.

U.S. Publication No. 2004/0210788 to William et al. discloses a method for testing synchronization between a first graphics processing unit coupled to a second graphics processing unit. The method includes detecting whether an incoming synchronization signal has been

received, determining whether the incoming synchronization signal is received from one of the first graphics processing unit, the second graphics processing unit and an external synchronization signal, and indicating on a control panel one of a first and second synchronization input/output ports on one of the first graphics processing unit and the second graphics processing unit as an input port and the other one of the first and second synchronization input/output ports as an output port, if the incoming synchronization signal is received from the one of the first graphics processing unit and the second graphics processing unit.

U.S. Publication No. 2004/0207618 to Williams et al. discloses a method for synchronizing two of more graphics processing units. The method includes the steps of determining whether the phase of a first timing signal of a first graphics processing unit and the phase of a second timing signal of a second graphics processing unit are synchronized, and adjusting the frequency of the first timing signal to the frequency of the second timing signal if the first timing signal and the second timing signal are not synchronized.

U.S. Publication No. 2004/0179019 to Sabella et al. discloses tile buffers in a graphics processing system are managed use "copy-on-write" semantics, in which tile data stored in a memory location is not transferred to another location until the tile data for one of the buffers is modified. Two memory spaces store tile data, and two logical buffers are used to access the memory spaces. For each tile, a tile association is maintained, indicating which of the two memory spaces is associated with each of the two logical buffers. To copy a tile of the first logical buffer to the second logical buffer, the tile association for the tile being copied is modified. Data for a tile is written to the memory space associated with a target logical buffer after ensuring that the tile association for the tile associates the target logical buffer with a different one of the two memory spaces from the other logical buffer.

U.S. Publication No. 2004/0169651 to Everitt et al. discloses lights that can be conservatively bounded within a depth range. When image pixels are outside of a light's depth range, an associated volume fragment does not have to be rendered. Depth bounds registers can be used to store minimum and maximum depth values for a light. As graphics hardware processes volume fragments overlapping the image, the image's depth values are compared with the values in the depth bounds register. If the image's depth is outside of the depth range for the light, stencil buffer and illumination operations for this volume fragment are bypassed. This optimization can be performed on a per-pixel basis, or simultaneously on a group of adjacent pixels. The depth bounds are calculated from the light, or from the intersection of the volume with one or more other features. A rendering application uses API functions to set the depth bounds for each light and to activate depth bounds checking.

U.S. Publication No. 2004/0153778 to Cheng discloses a system and method for configuring graphics processing communication among a graphics device, a chipset (a host bridge), and a data processor. A graphics driver is used to configure graphics communication within an information handling system using existing information stored in system memory or installing and running a configuration routine to determine a method of graphics communication. A configuration routine applies tests to determine a mode of data transfer between the system and the graphics device. Test results associated with the configuration routine are stored and can be loaded upon subsequent system startups to configure communications between the system and

the graphics device. A reliable mode for communicating between the graphics device and the information handling system is established to allow the graphics device to be used without requiring excessive interaction by a user.

U.S. Publication No. 2004/0036159 to Bruno discloses an integrated circuit having memory disposed thereon and a method of making thereof which includes a standard dimension carrier substrate and an information router integrated on the carrier substrate. Further included therein is at least one system memory integrated on the carrier substrate and in electrical communication with the information router across at least one of the electrical leads associated with the carrier substrate. Thereupon, system instructions may be stored and retrieved from the system memory through the information router within the integrated circuit on the standard dimension carrier substrate

U.S. Publication No. 2004/0012600 to Deering et al. discloses high-speed ring topology. In one embodiment, two base chip types are required: a "drawing" chip, LoopDraw, and an "interface" chip, LoopInterface. Each of these chips have a set of pins that supports an identical high speed point to point unidirectional input and output ring interconnect interface: the LoopLink. The LoopDraw chip uses additional pins to connect to several standard memories that form a high bandwidth local memory sub-system. The LoopInterface chip uses additional pins to support a high speed host computer host interface, at least one video output interface, and possibly also additional non-local interconnects to other LoopInterface chip(s).

U.S. Publication No. 2003/0212735 to Hicok et al. discloses a novel network architecture that integrates the functions of an internet protocol (IP) router into a network processing unit (NPU) that resides in a host computer's chipset such that the host computer's resources are perceived as separate network appliances. The NPU appears logically separate from the host computer even though, in one embodiment, it is sharing the same chip.

U.S. Publication No. 2003/0189565 to Lindholm et al. discloses a graphics hardware system and method for graphics processing. Such system includes a transform module positioned on a single semiconductor platform for transforming graphics data. Coupled to the transform module is a lighting module which is positioned on the single semiconductor platform for lighting the graphics data. Also included is a rasterizer coupled to the lighting module and positioned on the single semiconductor platform for rendering the graphics data. As an option, the graphics hardware system may further be equipped with skinning, swizzling and masking capabilities.

U.S. Publication No. 2003/0179220 to Dietrich, JR. et al. discloses a method and computer program product for generating a shader program. Initially, a file associated with a graphics effect is a selected. Such file is then read and processed. A shader program is subsequently generated based on the processing of the file to apply the graphics effect to an object.

U.S. Publication No. 2003/0151606 to Morein discloses a secondary representative Z value memory which includes a reduced-resolution representation of a primary representative Z value memory. Upon updating of a block of the primary representative Z value memory, one or more corresponding values are calculated for updating the reduced-resolution representation.

U.S. Publication No. 2003/0128197 to Turner et al. discloses systems and methods for supporting an external display on a portable device. A system on a chip (SOC) of the portable device provides a first set of graphics data to a graphics controller. The embedded graphics controller renders the first set of graphics data for output using an LCD screen integrated with the portable device. The SOC renders a second set of graphics data and provides rendered graphics data to an external display interface. The external display interface formats the rendered graphics data for output on an external, remote display.

U.S. Publication No. 2003/0112246 to Lindholm et al. discloses a system and method for a hardware implementation of a blending technique during graphics processing in a graphics pipeline. During processing in the pipeline, a plurality of matrices and a plurality of weight values are received. Also received is vertex data to be processed. A sum of a plurality of products may then be calculated by the multiplication of the vertex data, one of the matrices, and at least one of the weights.

U.S. Publication No. 2003/0112245 to Lindholm et al. discloses a graphics pipeline system and method for graphics processing. Such system includes a transform module positioned on a single semiconductor platform for transforming graphics data from object space to screen space. Coupled to the transform module is a lighting module which is positioned on the single semiconductor platform for lighting the graphics data. Also included is a rasterizer coupled to the lighting module and positioned on the single semiconductor platform for rendering the graphics data.

U.S. Publication No. 2003/0103054 to Montrym et al. discloses a graphics pipeline system and method for graphics processing. Such system includes a transform module adapted for receiving graphics data. The transform module serves to transform the graphics data from a first space to a second space. Coupled to the transform module is a lighting module which is positioned on the single semiconductor platform for performing lighting operations on the graphics data received from the transform module. Also included is a rasterizer coupled to the lighting module and positioned on the single semiconductor platform for rendering the graphics data received from the lighting module. During use, an antialiasing feature is implemented on the single semiconductor platform to improve a quality of the graphics rendering.

U.S. Publication No. 2003/0080959 to Morein discloses a method of graphics processing which includes determining a non-depth conditional status and an occlusion status of a fragment. Such a method may be used in culling occluded fragments before expending resources such as processing cycles and memory bus usage. In one example, a scratchpad stores depth values of robust fragments and is used for occlusion testing. Graphics architectures, and methods that include use of representative Z values, are also disclosed.

U.S. Publication No. 2003/0038808 to Lindholm et al. discloses a method, apparatus and article of manufacture for sequencing graphics processing in a transform or lighting operation. A plurality of mode bits are first received which are indicative of the status of a plurality of modes of process operations. A plurality of addresses are then identified in memory based on the mode bits. Such addresses are then accessed in the memory for retrieving code segments which each are adapted to carry out the process operations in accordance with the status of the modes. The

code segments are subsequently executed within a transform or lighting module for processing vertex data.

U.S. Publication No. 2003/0034975 to Lindholm et al. discloses a method and apparatus for a lighting system for graphics processing. Included is a plurality of input buffers adapted for being coupled to a transform system for receiving vertex data therefrom. The input buffers include a first input buffer, a second input buffer and a third input buffer. An input of the first buffer, the second input buffer and the third input buffer are coupled to an output of the transform system. Further included is a multiplication logic unit having a first input coupled to an output of the first input buffer and a second input coupled to an output of the second input buffer. An arithmetic logic unit has a first input coupled to an output of the second input buffer. The arithmetic logic unit further has a second input coupled to an output of the multiplication logic unit. An output of the arithmetic logic unit is coupled to the output of the lighting system. Next provided is a first register unit having an input coupled to the output of the arithmetic logic unit and an output coupled to the first input of the arithmetic logic unit. A second register unit has an input coupled to the output of the arithmetic logic unit. Also, such second register has an output coupled to the first input and the second input of the multiplication logic unit. A lighting logic unit is also provided having a first input coupled to the output of the arithmetic logic unit, a second input coupled to the output of the first input buffer, and an output coupled to the first input of the multiplication logic unit. Finally, memory is coupled to at least one of the inputs of the multiplication logic unit and the output of the arithmetic logic unit. The memory has stored therein a plurality of constants and variables for being used in conjunction with the input buffers, the multiplication logic unit, the arithmetic logic unit, the first register unit, the second register unit, and the lighting logic unit for processing the vertex data.

U.S. Publication No. 2003/0020720 to Lindholm et al. discloses a method, apparatus and article of manufacture for sequencing graphics processing in a transform or lighting operation. A plurality of mode bits are first received which are indicative of the status of a plurality of modes of process operations. A plurality of addresses are then identified in memory based on the mode bits. Such addresses are then accessed in the memory for retrieving code segments which each are adapted to carry out the process operations in accordance with the status of the modes. The code segments are subsequently executed within a transform or lighting module for processing vertex data.

U.S. Publication No. 2002/0196259 to Lindholm et al. discloses a graphics pipeline system and associated method for graphics processing. Such system includes a transform module adapted for receiving graphics data. The transform module serves to transform the graphics data from a first space to a second space. Coupled to the transform module is a lighting module which is positioned on the single semiconductor platform for lighting the graphics data. During use, the graphics pipeline system is capable of carrying out a fog and blending operation.

U.S. Publication No. 2002/0180740 to Lindholm et al. discloses a graphics pipeline system and associated method with an integrated clipping operation. First included is a transform module positioned on a single semiconductor platform for transforming graphics data from a first space to a second space. Also provided is a lighting module positioned on the same single semiconductor platform as the transform module. The lighting module is adapted for performing lighting operations on the graphics data. A clipping operation is also performed utilizing the

single semiconductor platform.

U.S. Publication No. 2002/0118308 to Dujmenovic discloses improved television tuning circuits. An example tuning circuit includes a fraction-N frequency synthesizer facilitating fine tuning. This tuning circuit may be formed using relatively few independent oscillators. The tuning circuit lends itself to the formation of an tuning circuit on an integrated circuit substrate. As well, this tuning circuit may be used to form a dual tuner tuning circuit integrated on a single integrated circuit substrate.

U.S. Publication No. 2002/0085007 to Nelson et al. discloses a method and computer graphics system capable of implementing multiple pipelines for the parallel processing of graphics data. For certain data, a requirement may exist that the data be processed in order. The graphics system may use a set of tokens to reliably switch between ordered and unordered data modes. Furthermore, the graphics system may be capable of super-sampling and performing real-time convolution. In one embodiment, the computer graphics system may comprise a graphics processor, a sample buffer, and a sample-to-pixel calculation unit. The graphics processor may be configured to receive graphics data and to generate a plurality of samples for each of a plurality of frames. The sample buffer, which is coupled to the graphics processor, may be configured to store the samples. The sample-to-pixel calculation unit is programmable to generate a plurality of output pixels by filtering the rendered samples using a filter.

U.S. Publication No. 2002/0015055 to Foran discloses a method and system for presenting three-dimensional computer graphics images using multiple graphics processing units. The dimensions of the scene to be rendered are bounded by a rectangular volume decomposed into rectangular subvolumes. Vertices of graphics primitives are compared with subvolume boundaries to determine to which subvolume a graphics primitive should be assigned. A GPU is assigned to each subvolume to render the graphics data that lies within it. A viewing position point is determined and communicated to each GPU. Rendered graphics data from each GPU are ordered based upon the viewing position Outputs of the individual GPUs are combined by blending within an image combiners. Outputs of image combiners can be presented for viewing or further combined in a subsequent stage image combiner.

U.S. Publication No. 2001/0029556 to Priem et al. discloses a direct memory access (DMA) circuit which is physically positioned with an input/output device, the DMA circuit storing a first reference value pointing to a data structure which describes a buffer portion of system memory in which data is stored for transfer to the I/O device, a value determining a position within the buffer portion of system memory beginning at which a next sequence of data is to be placed, and a value determining a position within the buffer portion of system memory from which a next sequences of data is to be copied to the I/O device, the DMA circuit including circuitry for reading data from the buffer portion of system memory beginning at the position from which a next sequences of data is to be copied and for writing the data read to the I/O device.

The "Go Multiple" PowerPoint Slide Show (10 Slides) presented by Dennis Yang at the 2003 SIS Platform Conference discusses the trend of graphics architectures becoming similar to CPU architecture, and suggesting that the graphics chip should be separated into two or more chips to have more room for performance advancement,...[just] like CPU has north bridge and

south bridge to help CPU to work with different parts inside a computer.

Scientific publication entitled "Chromium: A Stream-Processing Framework for Interactive Rendering on Clusters" by Greg Humphreys et al., describes Chromium, a system for manipulating streams of graphics API commands on clusters of workstations. As disclosed, Chromium's stream filters can be arranged to create sort-first and sort-last parallel graphics architectures that, in many cases, support the same applications while using only commodity graphics accelerators.

Scientific publication entitled "Hybrid Sort-First and Sort-Last Parallel Rendering with a Cluster of PCs" by Rudrajit Samanta et al., describes a hybrid sort-first and sort-last approach for parallel polygon rendering, using a cluster of PCs as a target platform. This approach performs view-dependent and coordinated partitioning of both the 3D model and the 2D image.

A separate listing of the above references on PTO Form 1449 and copies of all non U.S. patent references are enclosed herewith for the convenience of the Examiner.

Enclosed in payment of the requisite fee of $180.00 is Thomas J. Perkowski, Esq., P.C. Check No. 7027. The Commissioner is hereby authorized to charge any fee deficiencies or overpayments to Deposit Account No. 16-1340.

Respectfully submitted,

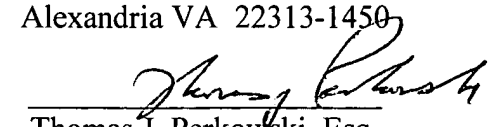
Dated: January 10, 2008

Thomas J. Perkowski
Reg. No. 33,134
Attorney for Applicant
Thomas J. Perkowski, Esq., P.C.
Soundview Plaza
1266 East Main Street
Stamford, Connecticut 06902
203-357-1950
http://www.tjpatlaw.com

CERTIFICATE OF MAILING UNDER
37 C.F.R. 1.08

I hereby certify that this correspondence
is being deposited with
the United States Postal Service
on January 10, 2008
in a Postage Prepaid envelope as
First Class Mail, addressed to:

Commissioner of Patents
P.O. Box 1450
Alexandria VA 22313-1450



Thomas J. Perkowski, Esq.
Reg. 33,134
Date: January 10, 2008